



Callbar Simulator Frontend

Diogo Pedro Barreira Lopes

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisors: Prof. Duarte Nuno Jardim Nunes
Prof. Sónia Isabel Ferreira dos Santos Rafael

Examination Committee

Chairperson: Prof. João António Madeiras Pereira
Supervisor: Prof. Duarte Nuno Jardim Nunes
Member of the Committee: Prof. Rui Filipe Lima Maranhão de Abreu

October 2019

Acknowledgments

I would like to thank my family for all their love and support over all these years, for always being there for me and directly or indirectly helped through the problems and obstacles that were appearing in my path.

I would also like to acknowledge my dissertation supervisors Duarte Nuno Jardim Nunes, Sónia Isabel Ferreira dos Santos Rafael and all the other assistant professors involved in this project that inevitably ended up at this thesis for their insight, support and sharing knowledge that helped me and my group colleagues made this project and thesis possible.

Since our project was made and developed for Talkdesk, I could not forget all those professionals and teams, especially Carlos Morgado and Fernanda Machado, that helped me and shared their knowledge and suggestions through the various meetings at their offices, physical and digital space given to us group at their facilities. Thank you all Talkdeskers that cross paths with me.

Last but not least, to all my friends and colleagues that helped me grow as an individual and were there for me during the good and bad times in my personal and academic life. Thank you.

To each and every one of you – Thank you.

Abstract

This thesis is inserted in the context of the IST-SCOPE project for the Master Thesis in Information Systems and Computer Engineering. It is a partnership between Instituto Superior Tecnico (IST), Faculdade de Belas Artes (FBA) and Talkdesk where we develop a product for the company in a team. Our group project consists in a real case simulation of Talkdesk's product (Callbar version) through a visual and interactive interface taking into account several concepts that are analysed in the real world like concurrency, scalability and simulate behaviour when possible. My personal main contribute and responsibility was the development of the front-end of the interface.

This document covers the IST-SCOPE project as well as the entities involved (IST, FBA, Talkdesk), the team and the phases of development of the group project but explaining in more detail my personal contribute – front-end development.

Keywords

Callbar, Simulator, Project, User Interface, Team, Research, Concurrency, Reliability, Scalability, Visualization.

Resumo

Esta tese está inserida no contexto do projecto IST-SCOPE para a Tese de Mestrado em Engenharia Informática e de Computadores. Trata-se de uma parceria entre o IST, a FBA e a Talkdesk onde desenvolvemos em grupo um produto para a empresa usar. O nosso projecto de grupo consiste em fazer uma simulação real do produto da Talkdesk (versão da Callbar) através de uma interface visual e interactiva tendo em atenção vários problemas que possam surgir no mundo real, como concorrência, escalabilidade e simulação de comportamento humano quando possível. O meu contributo pessoal e responsabilidade foi no desenvolvimento do front-end da interface.

Este documento inclui o projecto IST-SCOPE bem como as entidades envolvidas no projecto (IST, FBA, Talkdesk), a equipa (grupo) e as fases de desenvolvimento do projecto em grupo e a explicação em detalhe do meu contributo individual – desenvolvimento do front-end.

Palavras Chave

Callbar, Simulador, Projecto, Interface do Utilizador, Equipa, Concorrência, Fiabilidade, Escalabilidade, Visualização.

Contents

1	Introduction	1
1.1	IST-SCOPE	4
1.2	Project Description	5
1.3	Entities Involved	5
1.3.1	IST	5
1.3.2	FBA	6
1.3.3	Talkdesk	6
1.3.4	The Team - Capstoners	6
2	Motivation	9
2.1	The Company	11
2.1.1	Product	12
2.1.2	Values	14
2.1.3	Clients	15
2.1.4	Needs	15
2.2	Capstone Projects	16
2.3	Team and Individual Contributions	17
3	Research	21
3.1	Team Model Canvas	23
3.2	User Research	25
3.2.1	Meetings	26
3.2.2	Shadowing	27
3.2.3	Interviews	27
3.2.4	Scenarios	27
3.2.5	Behaviour	28
3.2.6	Personas and Profiles	28
3.3	Theoretical Concepts	28
3.3.1	Simulation	28

3.3.2	Automation	29
3.3.3	Testing	29
3.3.4	Metric Results and Analysis	31
3.3.5	Scalability	31
3.3.6	Reliability	31
3.3.7	Concurrency	32
3.3.8	User Interface	32
3.3.9	Messaging System	32
3.3.10	Presence System	33
3.4	Project Description and Scope Evolution	34
3.5	Methodology	38
3.6	Tools and Technologies	39
3.7	Roadmap and Schedule	39
4	Callbar Simulator	43
4.1	Business Model Canvas	45
4.2	Development Environment	48
4.3	Project Phases	48
4.3.1	User Research	48
4.3.1.A	Shadowing	49
4.3.1.B	Interviews	50
4.3.1.C	Meetings	50
4.3.2	Team and Project Management	50
4.3.3	Design	50
4.3.3.A	Interface Architecture	50
4.3.3.B	Team Website Architecture	50
4.3.3.C	Wireframes	51
4.3.4	Callbar Simulator Software Development	51
4.3.4.A	Front-end	51
4.3.4.B	Back-end	51
4.3.4.C	Integrations	51
4.3.5	Analysis and Report	52
4.3.6	Other Project Materials	52
4.4	Software Architecture	52
4.5	Back-end	54
4.5.1	Pseudo Presence System	54

4.5.2	Assignment	54
4.5.3	Variability and Profiles	55
4.6	Integrations	56
4.6.1	Talkdesk API	56
4.6.2	Protocols and Data Structures	56
4.7	Front-end	58
4.7.1	Visualization	59
4.7.2	Style	60
4.7.3	Interaction	61
4.7.4	Screens	61
4.8	Restrictions and Limitations	64
5	Conclusion	67
5.1	Results	70
5.2	Future Work	70
A	Shadowing Log	75
B	Interview Questions	79
C	Talkdesk API (endpoints example)	81
D	JSON File (example)	83

List of Figures

2.1	G2 Grid Scoring - Satisfaction/Market Presence.	12
3.1	Team Model Canvas Diagram.	23
3.2	The Testing Pyramid.	30
4.1	Business Model Canvas.	45
4.2	Project Phases.	49
4.3	Front-end architecture.	53
4.4	Finite State Machine for presence system.	55
4.5	Loading Screen.	62
4.6	Simulator First Screen.	62
4.7	Test Setup Preset Scenarios.	63
4.8	Test Setup Custom Scenario.	64
4.9	Callbar Simulator Main Screen.	65
4.10	Test Running Information.	66
C.1	Example of Talkdesk API - input and output.	82

Acronyms

AI	Artificial Intelligence
API	Application Programming Interface
AMQP	Advanced Message Queuing Protocol
BCBP	Browser Communications Backup Plan
BMC	Business Model Canvas
CCaaS	Contact Center as a Service
CD	Continuous Delivery
CEO	Chief Executive Officer
CPaaS	Communications Platform as a Service
CRM	Customer Relationship Management
CSS	Cascading Style Sheets
DOM	Document Object Model
FBA	Faculdade de Belas Artes
FSM	Finite State Machine
GCN	Global Communications Network
GLL	Global Low Latency
HCI	Human–Computer Interaction
HMI	Human-Machine Interface
HTML	Hyper Text Markup Language

HTTP	Hypertext Transfer Protocol
IoT	Internet of Things
IP	Internet Protocol
IST	Instituto Superior Tecnico
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
MQTT	Message Queuing Telemetry Transport
QA	Quality Assurance
QoS	Quality of Service
RP	Routing Platform
SaaS	Software as a Service
SLA	Service Level Agreement
SVG	Scalable Vector Graphics
TCP	Transport Control Protocol
TMC	Team Model Canvas
UDP	User Datagram Protocol
UI	User Interface
UR	User Research
UX	User Experience

1

Introduction

Contents

1.1	IST-SCOPE	4
1.2	Project Description	5
1.3	Entities Involved	5

The evolution of education in engineering schools has sought to bring the experience of students closer to the needs and challenges of the labor market. There are two main concepts that contribute for this new academic approach and this kind of projects – the interdisciplinary approach to solve even more complex problems and the approach to highly competitive global markets. These needs in the current days of academic research especially in engineering schools directly link to the main scope and goal of the developed project described in this thesis document.

My contribution was part of a **team work** and the result was a **product – Callbar Simulator**. The project is a Capstone Project inserted in IST-SCOPE (see section 1.1).

The developed work is an automated simulator that we consider as a product and we intended to evaluate and simulate the use of a target company's product – Callbar new version – from the company Talkdesk through a visual and interactive interface. My main personal contribute was the development of the front-end of the simulator (more information in section 2.3).

Although the product made was a group project developed in group composed by a team of four people this document report thesis is individual.

The formal name of the Master Project is **Callbar Simulator - IST-SCOPE - Talkdesk**.

It is an automated simulator that offers a visual and interactive interface that tests the Callbar product (the main theoretical concepts are detailed in section 3.3) of the company Talkdesk which is the company for which the work was developed and IST-SCOPE is part of the name of the multidisciplinary team responsible for the development of the project (more details in section 1.3.4).

1.1 IST-SCOPE

IST-SCOPE is a project included in an innovative pedagogical new project taking place at Instituto Superior Tecnico (IST)'s master projects. The project is developed by a group of students from different courses making it a multidisciplinary team.

This innovative project aims to approach students' experiences and studies to the enterprise world and to fulfill companies' needs. This way the companies get actual real complex engineering problems solved by interdisciplinary teams of students who acquire business competences as well as soft skills and increased tech stack.

The aim of this new academic model is to provide to the students ways and real tools to interact with the professional market and to give them options to do the thesis besides the traditional method.

The Scope model in IST is an innovative way to do the final thesis in the courses where the students in groups (usually three to five people) interact with a company to develop a product, add features to an existing product or service, or solve existing real problems. The interaction between the students and the company is supervised by a mentor which facilitates and helps in the organization of the project.

This particular SCOPE project introduces an innovative model of interdisciplinary work collaboration between various engineering fields and specialties at IST – Industrial Engineering and Management; Information Systems and Computer Engineering – and also with Faculdade de Belas Artes (FBA) – Communication Design and New Media. The work is developed through an academic group (team) of dissertation and project students further discussed in this thesis (see section 1.3.4) in order to achieve a common goal satisfying both worlds – academic and enterprise – solving real problems posed by companies in a practical way.

With this approach, not only I was able to work with my other three colleagues but also with the target company – Talkdesk – making sure that the academic side and research of the project fulfils the thesis requirements for this innovative academic project thesis.

IST-SCOPE follows the model of a Capstone project further discussed in section 2.2.

1.2 Project Description

One aspect of this project is that when it started, it was a completely blank page. Initially we were just a group of four students supervised by some professors with a target company – Talkdesk. No project, no specific goal, nothing more as opposed to the traditional thesis model (at least at IST) which usually has an objective and detailed description of the dissertation project, study and its goals.

In order to build up the project bases and the scope of the project and to come up with an objective description of the project, we (team) had to meet with the target company to meet each other, understand its values, strengths and weaknesses (detailed in section 2.1) so we could propose to develop or improve a project, product or service and then establish a bridge with the academic world to make the thesis.

This project description was a work in progress and therefore the scope of the project suffered some changes and tweaks throughout the time due to company requirements, academic expectations and external constraints and limitations.

The history and progression of the project's description and scope is described in section 3.4.

1.3 Entities Involved

The entities that are considered in this section are the ones with human assets directly related to the project – the team (students), the company (professionals) and the universities (professors).

1.3.1 IST

Instituto Superior Técnico is a school that aims to contribute to the development of society and people through sharing excellence in higher education in the fields of Engineering, Technology, Science and Architecture.

IST is humanly represented in this project by one supervisor professor, other assistant professors and three student elements of the team of two different courses – Industrial Engineering and Management; Information Systems and Computer Engineering.

This thesis was developed in the scope of my master's degree in Information Systems and Computer Engineering and therefore submitted and presented at IST. It was also at this university physical spaces that took place periodically scheduled meetings with the team, assistant professors and another team of a different project where progress and knowledge were shared.

1.3.2 FBA

Faculdade de Belas Artes is an artistic school that provides higher education creating professionals in the fields of Arts and Design.

FBA provides input in this project through the fields of design applied to technology and digital and it is represented by one supervisor, other assistant professors and one student element of the team – course of Communication Design and New Media.

Although this school was not the usual meeting point, it also served as a physical work environment in a later stage of the project.

1.3.3 Talkdesk

Talkdesk is the target company for which the project was developed and the working place we (team) met and worked most of the time. This company was the focus of our project and therefore the one we studied, analysed, addressed and treated as a mix of client and partner.

The research we made about Talkdesk's professionals, values, product, needs and more may be seen in section 2.1.

1.3.4 The Team - Capstoners

The team is a group of four university students finishing their master's dissertations – me and my three other colleagues.

The multidisciplinary team is composed by the following four people:

Carlos Silva Student from Faculdade de Belas Artes in Master in Communication Design and New Media - *UX/UI Designer*

Diogo Lopes Student from Instituto Superior Técnico in Master Degree in Information Systems and Computer Engineering - *Software Developer (Front-end)*

Miguel Pinto Student from Instituto Superior Técnico in Master Degree in Information Systems and Computer Engineering - *Software Developer (Back-end)*

Ricardo Monteiro Student from Instituto Superior Técnico in Master in Industrial Engineering and Management - *Project Manager*

Team members' responsibilities and individual contributions are detailed in section 2.3.

2

Motivation

Contents

2.1 The Company	11
2.2 Capstone Projects	16
2.3 Team and Individual Contributions	17

My personal research, work and development is included in the team's project and it is considered a product, more precisely an automated simulator that stresses out and tests the Talkdesk's product – Callbar.

Unlike the traditional thesis model and common dissertation researches, this capstone model incentives a more practical research in close relation with the various entities, especially the targeted company through meetings and other procedures (as detailed in section 2.2).

In this section is described the company's product, values, expectations, needs and their clients' demands, bearing in mind that this research and later the project development was relevant for that specific period of time, as digital and technological projects' life cycles may be very short and the company's work and development pace is much faster than normally the academic pace of projects and research or development projects is.

2.1 The Company

Created in 2011 by Tiago Paiva (still the current Chief Executive Officer (CEO)) and Cristina Fonseca, Talkdesk is a cloud-based contact center software provider in the digital industry of Software as a Service (SaaS).

Talkdesk's headquarters are based in San Francisco (United States of America) and with offices in four countries, in the cities of London, Lisbon, Porto, Coimbra, Aveiro, Madrid and South Lake City. Lisbon office was our main meeting point both with Talkdesk staff and for the product development as well.

Both Tiago Paiva and Cristina Fonseca are alumni of IST that realized the paradigm of contact centers was outdated and Talkdesk emerged from a contest hosted by a Communications Platform as a Service (CPaaS) company named Twilio in which they created a first version of Talkdesk using Twilio's Application Programming Interface (API) in just ten days. Then the startup was created and since those days, Talkdesk has been increasing its visibility, dimension and money at an impressive rate.

Since the beginning of this project (in September 2018) Talkdesk became a Unicorn and expanded its offices including research labs, staff and investment. Its products and services have nowadays acknowledged quality and not only the company is the youngest Contact Center as a Service (CCaaS) to be included in the Gartner MQ report but it also has at the moment one of the highest customer satisfaction scores according to G2 Grid as may be seen in figure 2.1.



Figure 2.1: G2 Grid Scoring - Satisfaction/Market Presence.

2.1.1 Product

Target company's products and services are where most of our attention was focused since the very beginning of the project because the main goal of the capstone project is to increase value and improve some needs of the company and improving the quality of its products and services is the right way to achieve success.

Talkdesk has many services all focused on its main product – **online cloud-based contact center software**. This contact center has many features as Artificial Intelligence (AI), automation, digital channels, reporting, analytics, management, integrations, voice, text and user experience. All these services get frequent changes and new versions come up regularly so the overall product is up-to-date with real-world requirements and customer expectations – so new versions of each service, product or feature require testing and analysis which is very useful and one focus of capstone projects which got our team's attention.

This main product allows the customer to exploit many advantages like one-hundred percent uptime

Service Level Agreement (SLA), global coverage and call quality, every layer of the product with AI, flexible deployment, security, workforce management, report and analytics, intelligent routing and endless integrations with already existing external services.

In a company like Talkdesk, new features, services and products are developed and new versions created in a fast pace and at the date of our research and beginning of our project the main platform services we choose to work on, focus and propose value are described below:

Real-Time Events - Automated processes by subscribing to real-time contact center events. Changes, messages, people status and activities are constantly occurring and interchanging. Agent (abstraction of people or bot using the contact center to communicate) presence status must change according to their actual state and availability at the moment.

Our project must fulfil these requirements and allow messages and events to flow in real-time to the interface simulator enabling visualization also in real-time.

Call Activity - All activity regarding agents and customers calls – inbound calls, outbound calls, ringing groups, answered and rejected calls, device connectivity from small quantities to massive amounts testing scalability and concurrency.

There are many features of Talkdesk's product regarding this topic and we should analyse and deal with the ones with highest priority for our project – as we will later see in this document in section 3.4, inbound calls, ringing groups, accepted and rejected calls and device states over time.

Agent Activity - Each agent (mostly human) has a presence tag according to his availability (occupied, available, after task work, temporarily unavailable, etc).

As may be later seen in section 3.3.10, presence system is very important in a real time service and in our project we consider this agent abstraction in the notion of device which has a unique state. So we consider each device to behave as an agent.

Callbar - Callbar is the main product of Talkdesk that we studied and it was the main focus of our project.

Callbar is the application that allows to receive and make calls to anywhere in the desktop and mobile phones increasing agent efficiency with web-based agent experience without even having the browser open. It requires no configuration thus it is easy and simple to use and to deploy on big companies with several workstations.

One of the features that make a huge advantage is the fact that it integrates with several Customer Relationship Management (CRM)'s and applications such as Salesforce, Zendesk, Bigcommerce, Google, Slack, Contactually and many more. This feature allows to access information that is stored in those CRM and add it to the Callbar to ease the process of searching for that information.

Some of the features present in this application are: inbound and outbound calls, transferring calls between agents (warm - with a previous talk to put the new agent in context or blind - transfer without talk), record the call and see contact information beforehand.

At the beginning of our project research, after meeting with Talkdesk employees our capstone goal and scope was to test a new version of Callbar with some more features and changes that should had been finished in the meantime (topic discussed in section 3.4).

2.1.2 Values

Talkdesk aims to empower companies continuously improving their customer experience and satisfaction with quality, reliability and security providing a cloud based contact center with enterprise-class performance and consumer-like experience allowing the service to adapt to the customers' expectations and team work flows increasing customer satisfaction, productivity and therefore profit. The company compromises to deliver a contact center platform with guaranteed reliability and one-hundred percent uptime SLA with voice quality and security of data.

Talkdesk focus on customer experience, reliability, data security, quality and therefore their clients' satisfaction, productivity and profit.

"In addition to features and scalability, Talkdesk provides solid security and customer support." – Talkdesk team.

In order to deliver the best security, quality and services Talkdesk invests in more technology and process, and therefore qualified people.

2.1.3 Clients

Talkdesk's clients are the product consumers and as it is a contact center they expect quick and personalized service constantly.

Whether serving patients, policyholders, students or shoppers, it is essential a comprehensive and fully integrated contact center solution. This solution helps clients accelerate business processes, increasing performance and customer experience through a cloud-based AI driven compilation of applications.

Talkdesk cloud-based data center is provided to different customers with different needs and scales from a small company or team with just a few employees to a colossal multi-national enterprise with tens of thousands of professionals with different devices all accessing Talkdesk's product at the same time and interchanging messages being voice or text.

2.1.4 Needs

Besides the common needs all companies have, especially tech companies, some aspects that Talkdesk tries to improve are the following:

Global Communications Network (GCN) - Talkdesk has strategically distributed data centers around the world and prioritizes redundancy on every level to eliminate single point of failure and therefore acquire industry-best reliability and global call quality.

Global Low Latency (GLL) Architecture - GLL ensures that each call travels the most direct and fastest route through regional communication nodes minimizing dropped packets and delay.

Browser Communications Backup Plan (BCBP) - If the telecommunications provider experiences issues (when external problems occur), BCBP ensures that the customer does not have downtime or call quality issues.

SLA - Talkdesk offers the industry's first and only one-hundred percent uptime SLA so any update in the product or change in the service must not interfere with this requirement.

Enterprise-Class Security - Talkdesk takes security and privacy very seriously and has implemented a variety of safeguards in design and code to carefully protect the clients' data.

Trust and Reliability - The company leverages modern technology to ensure that its products are

reliable, secure and always available.

Real-Time Updates - This aspect is useful not only when a problem event occurs keeping customers informed with relevant information and providing accurate timeline that estimates for issue resolution, but also real-time information on its services even when the normal flow proceeds.

After analysing these needs we tried to formulate a project proposal (section 3.4) that focused on reliability, scalability, concurrency, inducing entropy and induced problems.

2.2 Capstone Projects

Capstone projects are usually a two semester project at the end of the course and before entering the market place where students must work in group and independently research a topic. These assignments take significant planning to complete and the main objective is to challenge students to carry out a research and develop a project over a continued period of time for the students not only acquire knowledge but in the case of IST-SCOPE also assist a real company in some matter to be decided.

Capstone projects are also a good way for the students to demonstrate and increase their knowledge and skills and can be used as an employment portfolio and richer curriculum.

These paradigm of projects encourage students to develop their hard skills in their specialization field but also their soft skills like critical thinking, problem-solving engineering challenges, research, presentations and oral communication, document writing, teamwork and connecting students, academy and companies as well.

A generic guideline for a Capstone project model (especifically following IST-SCOPE) is:

1. Choose a target company and supervisor professor;
2. Meet the team – other group colleagues for the same capstone project;
3. Get to know the company – values, product, service, philosophy, needs, strengths and weaknesses;
4. Formulate a topic and a project proposal;
5. Build the foundations of the project – related work, investigation and research over theoretical and practical aspects;
6. Present the research findings and formulate a scope for practical development (or even dissertation) project;

7. Work in close relation between school and company.
8. Finally deliver and present the product or documents (project, code, reports, videos, presentations, papers, websites, etc) to the target company and to school.

2.3 Team and Individual Contributions

The project was developed in group and all the four members (section 1.3.4) contributed equally in terms of time and effort to the project work. All of us tried to be part of every stage of the project giving our personal input and learning things that may be off of our specific scope, for example me giving my opinion on design subjects and conduction user research interviews with users.

Besides each member worked at (and contributed to) almost every stage of the project, each member of the team adjusted its efforts and responsibilities according to the personal interests, expertise and academic field.

In general, the software developers were responsible for the development of front-end and back-end of the application, the designer member was responsible for the creation and visualization aspects of the user interface and the project manager had the responsibility of the team's planning and results analysis resulting from the application project.

My personal contribution towards this project was weighed in both time and effort invested and described in the following research and development lists.

Research – corresponding mostly to the thesis project (first semester):

- Workflow in Talkdesk's engineering teams and the development and testing methodologies they followed.
- Technological needs of the company and study of the related theoretical concepts (sections 2.1.4 and 3.3).
- Technical issues that may occur on a contact center and study how we may simulate them.
- Callbar research and study of how users interact with the product (section 3.2).
(Inquiring users and get their satisfactions, problems, ideas, behaviours, etc, so personas and scenarios may be generated accordingly).
- Analyse how our project's product (interface) can assist Talkdesk's needs in terms of functionalities, quality, visualization and interaction.
- Meetings with team members, professors and company teams.

- Develop research documents, reports, middle-project presentations and informative website.
- Project proposal and scope elaboration.

Development – corresponding to our product development and final thesis (second semester):

- Front-end development – main task and main responsibility (section 4.7).
(browser application screens; visualization blocks; real-time graphic from a specific library; setup and custom feature; device activity filtering; simulator aspect, visualization and interaction).
- Integration between front-end and back-end following certain protocols (section 4.6).
(JavaScript Object Notation (JSON) files with relevant information; setup variables; customization values; default files of variables, scenarios and profiles).
- Retrieve/send data from/to back-end.
Picking values from the back-end simulator. Store and show values/metrics. Enable values to be viewed.
My contribution is aligned with the team (more specifically with the other software developer) to develop ways and endpoints to enable information to be viewed by the user and then be integrated in the user interface.
- Stilization of visual aspects according to material from the design stage (section 4.7.2).
- Test of the application (front-end side).
- Presentation materials development – presentation, promotional videos, thesis document and information website.

Besides the previous listing of my main contribution, I also invest some time and effort (although more equally divided with my team mates) on the following:

- **Analysis and Reporting**

Analysis of the metrics, generation of Key Performance Indicator (KPI)s,

My individual contribution is for support in this field and getting overall knowledge. Then test application times.

- **User Interface (User Interface (UI)) Design**

Design of the interface, adjustments in wireframes, non-function and functional prototypes.

My individual contribution is to provide ideas and input creativity for the model and its architecture.

- **Messaging System (back-end)**

Broker, Internet protocols, Internet of Things (IoT), messaging system, network, integration with Talkdesk API.

My individual contribution is to develop the model and system according to the protocols. Although, it is not my main focus and other software developer team member is the responsible one for this phase as this is back-end.

- **Communication**

Soft skills. General team work. Connection between professors, university and Talkdesk. Pitches. Presentations. Meetings.

My contribution was the same as for every team member. Each team member had the same amount of workload. Usage of Git for version control, Slack for communication with the company, email for communication with professors and other messaging tools for team work.

3

Research

Contents

3.1 Team Model Canvas	23
3.2 User Research	25
3.3 Theoretical Concepts	28
3.4 Project Description and Scope Evolution	34
3.5 Methodology	38
3.6 Tools and Technologies	39
3.7 Roadmap and Schedule	39

In this chapter it is described the theoretical concepts I analysed for the development of the project, some relevant technicalities, the user research performed by the team and finally the project description, technologies and tools used, scope evolution and its planning.

This research was mainly performed in the first semester as it founded the bases of the project although we continued researching even when developing the simulator because new requirements and features were arising.

Note that this project was very ephemeral as it was a long academic project and ambitious in terms of education but very short in terms of company needs. In addition to that some issues contributed to project changes and therefore the evolution of the scope as discussed in section 3.4.

3.1 Team Model Canvas

Team Model Canvas (TMC) is a Business Model Canvas (BMC) for teamwork in a form of a diagram that was built in group by the four members of the team with the goal of improving team work, resolve conflicts, bring members to the same page and therefore improve project development.

It is visible in figure 3.1 and also at [our site](#).

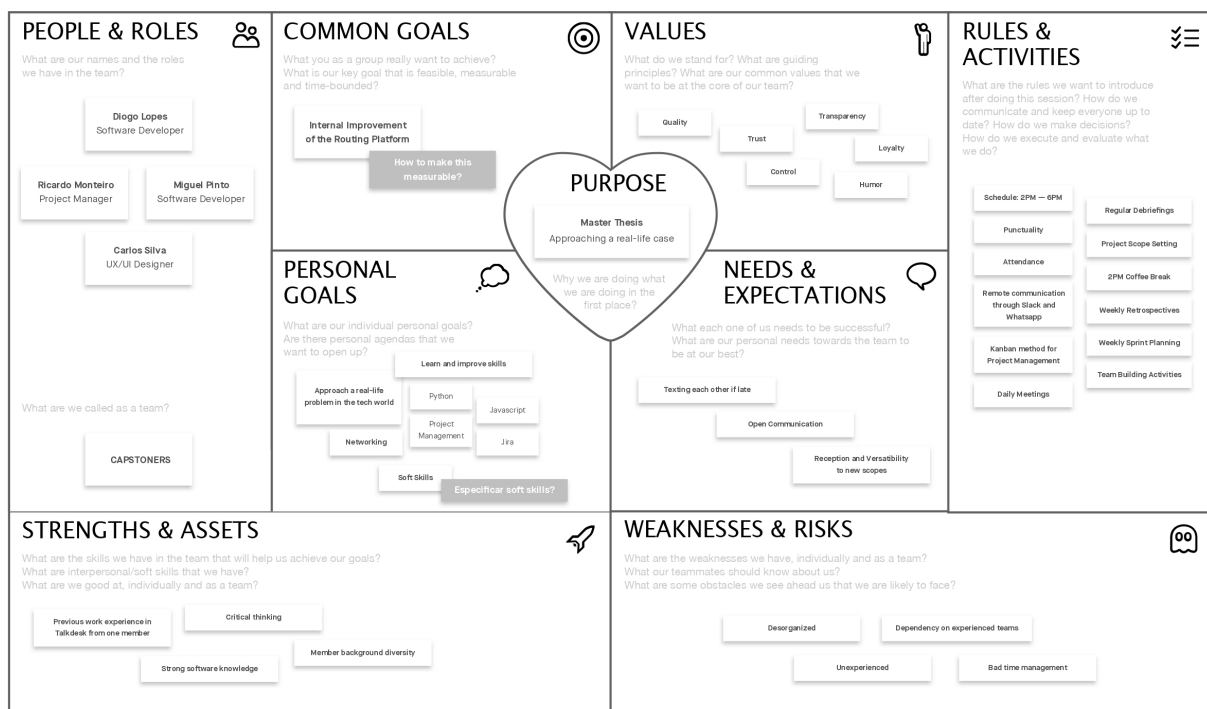


Figure 3.1: Team Model Canvas Diagram.

People and Roles

This group has already been described in section 1.3.4. Our team name is Capstoners because we did a capstone project.

Common Goals

Our common goal is to increase value and improve Talkdesk's product, more precisely a new version of Callbar and then a module of this product – Routing Platform – because of the scope evolution explained in section 3.4.

Personal Goals

The Personal Goals block is an aggregation of some of each team member's personal goals such as approaching a real problem case in a technological company; networking with other students, professors and company employees; learn and improve soft skills and tech skills (Python, JavaScript, project management, Jira software, communication).

Values

In this block it is described the guiding principles of the team. The team stands for project quality, loyalty, humor and good working environment, control and transparency of the developed simulator and trust.

Rules and Activities

The team followed some self-proposed rules and defined activities to make it easier to develop the project and to make the team-work more efficient, like a work methodology. To do that we scheduled team work at the company office every day from 2PM to 6PM and it was very important to be punctual and attend these meetings. Outside this schedule, we maintained a regular remote communication via Slack and WhatsApp. Other rules and activities were: following Kanban (Agile as described in section 3.5) methodology, having daily debriefings and meetings, setting project scope and continuously upgrade it (when necessary), weekly retrospectives and sprint planning, and also 2PM coffee break talk and team-building activities.

Needs and Expectations

In order to achieve success in our capstone project we needed to satisfy our personal needs being them texting or calling each other to talk about important matters of the project even if late in the day; open communication – no question is a dumb question; Reception and Versatility to change and adapt to continuous new scopes.

Strengths and Assets

Being this team a multidisciplinary one, the global strengths and assets have a larger range than the common academic projects usually have and our main advantages were: previous work experience in the company, critical thinking, strong software knowledge and background diversity and speciality.

Weaknesses and Risks

Because every team has its weaknesses, the capstoners were no exception. Our main disadvantages were disorganization, being dependent of other teams (for example the Talkdesk's product team that should have released the new version of Callbar for the team to test and simulate it), we were an inexperienced team basically working as a junior company team without senior supervision and guidelines and we also suffered from bad time management.

Purpose

The purpose of our team is what we were doing in the first place. We did this project to finalize our master thesis (in each member's course) by approaching a real-life case at a real tech company.

3.2 User Research

User Research is the investigation of some user characteristics to collect useful information. This information could be results of user testing, retrieving their behaviours, etc. It is used in many steps of product development and design and is easily used in Continuous Delivery (CD) and testing as well as agile methodologies (all this performed in our project and discussed in section 3.5).

There are many types of user research being them experimental or observational, qualitative or quantitative and are very well suited for multidisciplinary teams such as ours.

User Experience (UX) is the user's satisfaction, its emotions and attitudes about using a product or service. It combines practical and emotional important singularities and aspects of Human–Computer Interaction (HCI).

The main goal of UX is to make a product better. The quality of a product increases when its functionality performs well, is useful, easy to use and makes users happy.

UX and Research include the investigation done by us to the people that actually use the application, Callbar, to understand what is their typical day of work and what are their reactions to certain chains of events. In this user research we intended to do shadowing, to create personas and scenarios.

For example, with the shadowing and informal interviews we were able to create more realistic sce-

narios and adapt the personas to a real-world environment. It also helped us define the different test cases and simulation paths that we needed to cover for our simulator.

3.2.1 Meetings

It it consider a meeting whenever a team member meets with a Talkdesk employee (or any other human target to do research) being it a direct personal or remote meeting or even a small informal talk at the office regarding project and user research.

During the first semester I had almost every week meetings with the team members and Talkdesk's teams as well where we had the opportunity to meet them and have a bit of knowledge of what they do during their work. The teams that were presented to us were: **Callbar** (product), **Sales**, **Presence System** and **Cobalt** (design).

Callbar

This team is responsible to develop the application that all agents will use to communicate in the call center that uses the Talkdesk application. It was stated to us that our work would focus in the new version that would be released. There was no application deployed and our tests would need mocks to test the system as well as data to fill the entities of the agents.

Sales

In this meeting it was presented to us how the sales team sells the Talkdesk products to their potential clients and what were the greater pillars in the presentation to them (in accordance with the values of section 2.1.2): easy to implement, easy to use, it is a platform that was made to be configurable and not programmed in the client side and at last the administrator configures the tool and the supervisor analyses the KPIs.

Presence System

This team and the Callbar were our main focus to develop our test harness for the initial scope. This session allowed us to know how the system was before and how the new one works. The new presence system would be used in our project for the agent status (and their presence).

Cobalt

The Cobalt team is responsible to make the bridge between the UX/UI and the engineering team. They are responsible to create the components that standardize all the products interface that are presented

to the clients.

3.2.2 Shadowing

Shadowing is a qualitative research technique implemented in User Research where some research (in our specific case each one of team members) closely observes and accompany the work of some target user (analogous the user would be Callbar's agents or Talkdesk's support team workers) to acquire knowledge of how they use their product (Callbar) within their natural environment (offices) and understand their choices, problem-solving skills and (if possible) gathering information about their personalities in order to migrate that information to personas and therefore include real life-based knowledge into complexity of the simulation.

Shadowing allows us to understand existing behaviours of agents that use some product or service.

3.2.3 Interviews

An interview is a conversational meeting between two or more people where questions are asked and answers are given. It could be in formal or informal way.

When the team did user research no interview was in a formal way because Talkdesk's office environment and its employees are quite informal but some meeting interviews were in a private meeting room in a more traditional way and other interviews occur right in the coffee break area or in the elevator and felt more like a two way conversation than an interview. Nevertheless our goal was fulfilled on both cases as good information arise.

3.2.4 Scenarios

Scenarios could be a specific story within the user research scope that contribute to personas performances or some larger scale story to be integrated in various paths of our simulation that in our case includes different personas (with different behaviours, same goals, different characteristics, problem-solving skills, emotions, etc), different physical (hardware) issues and different number of agents as well (from just a few to potential millions). With this (two types of) scenarios it is possible to test the product in terms of scalability, concurrency and reliability all based in possible real cases (in accordance with Talkdesk's needs as discussed in section 2.1.4).

Many scenarios may be covered in the simulator in order to increase the approximation to the real-world scenario of the product use and for different numbers of agents and devices connected, as it already happens in various cases of Talkdesk's clients. Our simulator may have already existing (default) scenarios or the option for the user to setup his own kind of scenario for the simulator to run it.

3.2.5 Behaviour

Behaviour is a range of actions and decisions made by individuals. In our project we studied human agents to gather their behaviour and then replicate those actions and decisions to the automated simulation inherent to personas.

3.2.6 Personas and Profiles

Personas are fictional characters created by us that represent general agents. These personas are created based on user research and represent a group of users (Callbar agents) with similar behaviours and patterns using the product and reacting to some triggers.

In our project personas are agents that are represented in the notion of devices. All agents have a goal using Callbar but all goals will be the same for simplicity and would not be relevant for our scope otherwise. This common goal is just *receive calls*, *accept pending calls* when possible and *wait* for more calls. Besides personas' characteristics and goals they have an inherent scenario based on their behaviour that is acknowledged by some user research method.

In our project we considered Profiles as a representation of some user (or persona) characteristics. It is where variability is introduced so different representations of users may have different behaviours (detailed in section 4.5.3).

3.3 Theoretical Concepts

In this section I will cover and describe the most important theoretical concepts I had to study so I could perform my individual tasks towards the project elaboration and development.

3.3.1 Simulation

In computer science, simulation is the replication of a real-world system over time using a computer or a computation application (for example an interface) so it is very useful for engineering because simulation is a problem-solving powerful tool.

A simulation provides a virtual environment to learn and experiment some system, product or service where data is given as input and data is delivered as output and mistakes are deliberately induced so the simulation may convert them to improvement.

It is common to have a model of a simulation. The model represents the description of the real system – could be just a sample of it.

A simulation may give dynamic responses as output as it is a representation (although potentially simpler than the real-world) of real product usage and help determining behaviours when variables of

the system are modified. By simulating a system one may also test it in terms of dimension, scale, scenarios and behaviour.

3.3.2 Automation

Automation is used in a number of areas such as manufacturing, transport, utilities, defense, facilities, operations and information technology. In the information technology and computer science domain, a software code script may test a product and produce useful results that can later be analysed.

Automation in computer science is the creation of technology that allows a system to be performed autonomously without human intervention and assistance automatically interchanging digital information with computational power – in our case the system is Talkdesk’s product Callbar and it should run without human agents interacting with it as opposed with the real-world.

The benefit of automation includes company profit and savings (in time, electricity costs, material costs, employees, and so on) and improves quality, accuracy, and precision in most of the cases.

3.3.3 Testing

Software Testing is a way of gathering useful information (some results) regarding a product (software) functionality to later be delivered and analyzed either by quality assurance teams or development teams in order to increase the quality and performance of the product.

In our project the product is Callbar as previous discussed and the main goal of testing it was be to overload the system in terms of requests, differences of usage and inducing real-world problems into equation.

A *test-harness* is an automated test framework that includes a collection of software and test data configured to test a program unit by running it under many different scenarios and conditions and monitoring its behaviour and outputs. These outputs are important metrics that may be used to analyse the system performance. Test-harnesses particularly facilitate integration tests that was initially the main testing focus of our project. They have several advantages in terms of automation, overall system operation and massive data inputs. Therefore it is very well suited to study concurrency, scalability and reliability.

The collection of tests should be able to test the best case, intermediate complexity cases and worst-case scenarios as well for the various problems that may occur.

Black Box Approach

Since we had no knowledge of internal Callbar implementation and no access to the product source code it is considered a model of black box as we just had the product output (through API endpoints).

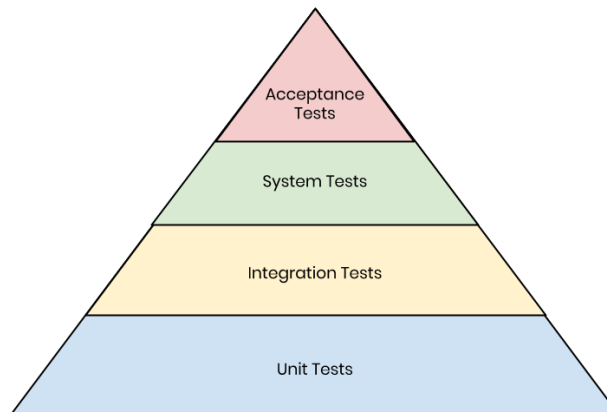


Figure 3.2: The Testing Pyramid.

Testing Pyramid

The Testing Pyramid principle says that a test-harness should have a lot of unit tests, many (but fewer) integration tests, some system tests and as few acceptance tests as possible, as may be seen in Figure 3.2.

- **Unit Tests** refer to the verification of operations and particular functionalities. In our case it would test specific connections and made Hypertext Transfer Protocol (HTTP) requests (such as *POST*, *GET* and *PATCH*). Although some static Callbar functionalities might be tested, it is more interesting to study and test data-flow analysis.
- **Integration Tests** aim to verify interface API connections that a test-harness can intensively stress.
- **System Tests** may use some of the previous tests to perform overall simulations of the system under bad conditions that may be induced.
- **Acceptance Tests** also known as operational readiness testing are performed in product (Callbar) pre-release. This software testing ensures the portability of the system making sure it is working as expected. If too much problems were induced, Callbar might have stopped its working leaving the overall states in unknown bad conditions and operating environment inoperative.

unittest Library

unittest is a Python's standard automated library for software testing. This library covers all types of tests present in the testing pyramid principle and it was our first choice giving the fact that Python was the main programming language chosen for this project development back-end.

3.3.4 Metric Results and Analysis

In our project metrics are considered software metrics and they are the output of our simulation and the results of the tests.

Metrics are properties of some software system that indicate values which may be analysed such as times measurements, number of calls, devices, issues reported, occupancy rate, satisfaction, to name a few examples in our scope.

3.3.5 Scalability

In the computer science domain, scalability could be analysed and discussed in depths by a program efficiency and its algorithms classification (for example the big O notation or asymptotic notation) but in this document I refer to scalability in a higher level and it is just the capability of a system to increase (or handling growing) large amounts of data.

A system is called scalable if it is possible to adapt and normally perform its function when adding more and more resources to the worst case point without failing or decreasing significant quality.

For an application to be scalable it needs to handle a growing amount of work. In this project the agents would input the growing amount of calls and with a call generator script we would be able to simulate incoming calls for the agents.

The dimensions of scalability directly studied and analysed in our project were:

Functional scalability - enhance the system adding new functionalities (and features) and still correctly perform the previous existing ones.

Load scalability - the system ability to expand and reduce to hold heavier or lighter loads of data – stress testing.

Administrative scalability - the ability for an increasing number of users or agents to access the simulator system (although in the abstraction of device and not agent).

3.3.6 Reliability

Reliability might have different meanings but in this project it is considered the engineering reliability. It is the ability of a product, service or system to correctly perform its required functions and satisfying its goals under the induced scenarios and therefore simulation results for a specified time.

We consider a product reliable if it well performs its purpose and fulfils the satisfaction requirements of the company regarding time usage and therefore the company's customer expectations (as discussed in section 2.1.3).

The system is considered reliable when it can handle a certain volume of work for a period of time without having failures (*"Our enterprise contact center platform delivers guaranteed reliability [...] to ensure the security of your data [...]"*). As a client, what is intended is 100% of up-time, which is a very difficult thing to accomplish as an infrastructure becomes increasingly more complex and developers must deal with strict time schedules for project completion.

3.3.7 Concurrency

Concurrency is the capability of a program (or in this case the product Callbar) to be used by many devices potentially at the same time by parallel execution and in no specific order sharing its resources and yet be able to perform its regular functions not affecting the final outcome.

3.3.8 User Interface

UI is the territory where humans and machines mingle. It is where interactions happen both from human side to machine side and otherwise – the human computer interaction.

Our project simulator is a simple example of a Human-Machine Interface (HMI) as it involves for the human user that will be the person using the simulator to setup the environment and then also interact with the visualization aspects and filtering features of the simulator interface and on the other side the simulator (machine) should output the results in a simple and user friendly way given the human input or a default scenario in the beginning of its operation.

3.3.9 Messaging System

Message Queuing Telemetry Transport (MQTT) is a lightweight protocol to sensors and small devices optimized to Transport Control Protocol (TCP) over Internet Protocol (IP) networks that are not reliable and have high latency. This protocol is based on the publish-subscribe model.

The publish-subscribe model allows the client that sends the message (publisher) to be decoupled from the one that receives it (subscriber). In the core of this model there are topics that the clients are subscribed. These topics will serve as routing rules to know to which clients to send the messages. The client can always add or remove more topics to control the messages that will receive.

Since this model decouples the publish from the subscriber it needs to have an intermediary piece to handle the messaging system - *broker*. The client can be any device that connects to this broker over a MQTT library on the network.

The broker is responsible to receive the messages and apply the filters to determine who is subscribed to each topic and send the corresponding messages to them. The broker also keeps track of the

persisted clients and the messages that have been sent and lost to apply the Quality of Service (QoS) policies.

QoS can be divided in 3 groups:

1. QoS 1 - at most once delivery, the broker won't give guaranties that the message was received, similar to User Datagram Protocol (UDP).
2. QoS 2 - at least once delivery, the broker guaranties that at least one message will arrive, similar to TCP.
3. QoS 3 - exactly once, the broker will guaranty that the message will be delivered exactly one time.

The broker is also responsible for the client's authentication and authorization.

3.3.10 Presence System

This is the management system where the presence of entities such as user, person or business is managed. It usually represents current attributes of the entities that should dynamically change in real-time. In our simulation, different devices have their states changing like it happens in the real-world as our devices may change from available to occupied, may ring or perform any other task and when a change occurs the system must change its behaviour and thus the presence system be instantly modified. Callbar and the new version also have a presence system (although with an upgrade in the new version) that should be accessed through API endpoints.

The presence system was still in development by Talkdesk for the new version (and was not ready for use as described in section 3.4) and it would be used as the new infrastructure for the agent status in the application that would be released.

This system is a core piece since it would be through it that is possible to access or modify the status and occupancy of agents, or subscribe to notifications whenever there is a change in presence.

With this system it would be possible to make the following operations:

- Retrieving a single agent presence status (any agent for the given account).
- Subscribing notifications for an agent's presence status changes.
- Subscribing notifications for the presence status change of all agents of an account.
- Updating any single agent's (including oneself) status.
- Modifying agent's occupancy.

3.4 Project Description and Scope Evolution

Our project is a capstone project that aims to improve a company's product or service. Therefore we had an inevitable dependency on Talkdesk's teams that are responsible for the product development that we wanted to test and simulate.

This product is Callbar and was already being commercialized to company customers so our analysis and simulator output would be much more useful and valuable if applicable to the new (and complete) version of Callbar. It is useful to compare a new version of Callbar to previous ones in terms of times, features, efficiency and efficacy – and we contributed exactly to stressing the system and testing its concurrency potentiality, scalability, efficiency and reliability to analyse if it was possible to expand the product to other potential bigger clients with different and more demanding requirements always maintaining the product's quality – to understand if there were improvements and in which conditions those occurred.

The new version of Callbar was planned to be finished at the starting time of our project development but unfortunately that didn't happen which delayed our development and to get things worst we had no guarantees of the estimated time delivery of this new version of Callbar (could be a week, two weeks, a month or even a year) and so the capstoners team first decided to wait some time and continuously seek updates of Talkdesk responsible people for the new version and then dynamically decide the next steps and adapt the scope.

Time had run and we still had no final, complete and stable version of the company's product to test and simulate – this matter was something out of our control. Given this issue we started to think about scope alternatives or even project change. After meeting with the company employees we were told that it was possible for them to give us not a complete version of the product Callbar but a simpler (and very important) core module of the product – **the Routing Platform (RP)**.

Instead of Callbar, at this moment our target product to be simulated was the RP which led to a scope update and evolution – testing the product in the terms of a test-harness (section 3.3.3) was no longer relevant but this scope evolved to a more generic simulation of the system and it would still be helpful for concurrency, scalability and reliability analysis.

RP is one of many modules that compose Callbar and it is one of the most important and core ones. It routes communication (calls) between devices through an API with given endpoints and returns a list of devices with their current states and values. It abstracts a representation of calls and device state changes which is a core functionality of a contact center. So testing and stressing it (scalability, reliability, concurrency, etc) according to the company needs (section 2.1.4) would increase this module value and therefore strengthen the future new version of Talkdesk's product Callbar.

After one of the first meetings at Talkdesk a first project description was shared by them aligned with the company's needs and expectations – the bases Talkdesk wanted for us to follow:

"The Capstone Test Harness for IoT Messaging System intends to operate an IoT system for tens of thousands of devices with strict latency requirements. The key factors in these systems are reliability and scalability to ensure they are able to accommodate the production load. This project aims to build a test harness capable of reproducing realistic loads. To this end the system has to be able to simulate tens of thousands of devices under network conditions, degradation and temporary loss of signal, etc."

Below are two different project descriptions representing the first one regarding the expected use of the final and complete Callbar and then the second one regarding the new product for us – RP.

First Project Description - at the time of project research (first semester)

The team was committed to develop a project that would increase the performance of Talkdesk product named Callbar which is an online cloud-based data center service that still had some restrictions and was exponentially being increasing in terms of users and therefore operations performed.

Talkdesk's service operates an IoT (Internet of Things) messaging system for many devices with strict latency requirements following some messaging protocols. So this project directly affected the promise the company sells to their customers which is *"Our enterprise contact center platform delivers guaranteed reliability [...] to ensure the security of your data [...]"*.

The "IST-SCOPE - Talkdesk" team would develop a solid and automated test-harness that would simulate agents (data-center operatives) and interactions within Callbar for different number of agents regarding cases (scenarios) from small teams (few agents) to enormous teams (millions of agents) applying for every case real-world problems that may occur like internet connection loss, signal loss, slow network, hardware issues in agents' headsets and clients' phones, agents and users (bad) behaviour, etc. This way would be tested the Callbar response, scalability from the number of IoT devices, reliability of the system that should not fail and retrieving relevant metrics from the automated tests like times, speeds, data flowing, number of problems that occurred, problems resolved, user satisfaction, user behaviour and agent's problem-solving skills, etc. After all the metric results were gathered for every case

it would be time to analyze these data results and to show them in a friendly user interface to Talkdesk people.

It was important the Test Harness results were reliable in order to analyze them and retrieve useful information to generate KPIs and core metrics guiding for a good solution and therefore Callbar improvement.

It would be also needed to generate different personas based on agents and users behaviour to determine various personalities of agents with different behaviours. To do so this project also included shadowing and other user research activities ideally performed in Talkdesk clients that use the service or inside the company (which also uses its own product).

In a more simplistic way the project may be divided in four stages.

Test Harness is the automated battery of tests to be performed in Callbar – different number of devices, various problems, operated between agents and end users all with different personalities and therefore different behaviours dealing with problem-solving situations. The Test Harness returns metrics.

Metric Results are the measurements retrieved from the tests that show the performance of the system to later be analyzed in order to gather useful information.

Result Analysis is the inherent knowledge that comes from the metrics. Used by data analysts to improve performance and businesses.

User Interface is a simple and friendly way to see information and metrics data. To be accessed by Talkdesk people either tech, engineers or regular non-tech people.

The main goal of this project would be to test and measure the performance of a Talkdesk application – Callbar.

Since we wanted to measure the performance we would need to stress the application with a well defined workload to increase the operations performed and the load of the application.

To better understand the scope of Callbar we can imagine that the service provided by this application can go from small companies that have fewer people to colossal multi-national enterprises with tens of thousands of employees with different devices all accessing the service at the same time and interchanging messages being voice or text. This brings the fact that it is possible that the application may become overloaded and depending on its policies it can start to trash the requests.

Some issues that might be analysed were:

- Large number of *logins* can cause data transfer failures between platforms.
- The problem of having to handle with huge queues with calls.
- Loss of data that was relevant to give the context awareness for the agent before answering the call. This was an issue that the Callbar had. When the information did not arrive the application would freeze without allowing to answer, reject or close.
- How agents behave when there is a problem or what is their normal behaviour when using the application.
- Data transfer interruptions as higher latency, temporary signal loss and agent errors.
- The size of the messages that may have impact on the performance of the broker.

Second and Final Project Description - After scope modification (second semester at development phase)

Because of the scope evolution our main goal for the project was no longer to perform a test-harness but a simulation instead, through a user-friendly interface.

The project is a visual and simple web interface to be deployed in the browser (ideally on Chrome as it is also the browser that Talkdesk offers support regarding its products) in a single page with the following features:

Setup Scenarios - The user may choose a scenario for the simulation to run or customize his own. Here it may be added some variability through the selection of profiles corresponding to personas and their behaviour (for instance a device with a "champion" profile means that the agent using that device will have a higher probability of answering a ringing call than an agent that has a "shy" profile.

Graphic Timeline - A real-time graphic that enables the visualization in real-time of calls being answered and rejected.

Device Activity - A visual block that also may have interaction. It contains all devices that were generated (from just a few to massive quantities) and the user may filter this visualization by the device's current state ("on call", "waiting for call", "ringing", etc) – also displayed in real-time whenever a change occurred (that we called an event).

Test/Simulation View - From the start of the simulation until its end it is possible to see the time elapsed and other useful simulation information like the number of devices and the ones already

answered, stop or reload simulation options.

For this development the team worked together mostly at Talkdesk's office in Lisbon using the best qualities of each member. My main and individual contribution was to develop the front-end of the interface following the design inputs and communication with back-end with *socket.IO* technology using our own data protocol exchanging *JSON* files with the relevant information we wanted to pass (back to front end and also front to back end) through endpoints.

Besides our main project being at this point the Callbar Simulator (which we consider also as a product) we also had to develop some palpable material such as an information website of the project (visible here at [our site](#)), promotional and informational videos and a final presentation of the product both to the academy and the company. My individual input in this tasks was the presentation elaboration (alongside my colleagues) and the website development that only had information to the deployed and therefore only had front-end).

3.5 Methodology

To the development of this project we intended to follow the Agile guidelines to help us on the iterative process. This kind of practice is adopted whenever there is a need to deliver high quality work with a fast pace (rapid delivery).

Scrum is a subset of Agile and a lightweight process framework to put in practice the guidelines for rapid delivery. The divisions of work are small (thus lightweight) and the process is called the Sprint (the development cycle) which may be a week, half a week, two weeks, it is the team discretion.

Although we (team) first had in mind to follow Scrum we decided to follow Kanban instead, after a meeting with a Agile Coach of Talkdesk and also after the scope change and therefore the planning alteration because we had less time for the development and the deadlines pressed us.

Kanban is also a subset of Agile. It is a framework used to implement agile software development requiring constant communication and team transparency while working – it just needs a Kanban board (could be a physical board or a digital one like we used in Jira software). Its purpose is to categorize all stages of the work in progress from the first left column of the Kanban board work flow (which usually is the backlog) to the right column (that could be the testing, implementation or delivery stage) of each work iteration or version. So we choose Kanban because we had less time than we expected (as discussed in section 3.4) and it is basically just moving task cards from one board column to another until that task is completed and finally all the cards would be done which means that the project (or at least that project

phase) was completed.

3.6 Tools and Technologies

In this section I will mention the tools and technologies I personally used or researched (not the ones other colleagues might have used, for example the design software for image or video editing since it was not a task that I contributed).

As described in the section 3.4, our scope suffered some modifications and we focused more on the simulation than the initial idea of an automated test-harness. Nevertheless, not only I did research about it but also developed the bases of the tests to be performed in Python using the unittest library.

All the back-end was developed in Python with the help of some tools such as Pika for the implementation of Advanced Message Queuing Protocol (AMQP) protocol libraries as *request*.

For the front-end and also our project information website I used web development technologies such as HTML5, CSS, JavaScript Vanilla and also D3.js for the real time visualization graphic (detailed in section 4.7).

Version Control – Git

During the development phase we used git to maintain control on our versioning of the code that was being created.

Since we used real endpoints and connectivity with the Talkdesk's API and its product, some information is not to be shared and therefore our git repository is private and not able to be shared – the people with access were just the team members and some Talkdesk employee if necessary.

3.7 Roadmap and Schedule

We named the project planning by *Roadmap* and it is goal oriented.

Both Schedule and planning suffered some modifications due to the scope evolution but the main aspect remained intact. Our time planning changed mainly because of the team's lack of experience in this kind of projects, knowledge of the company work pace and also to external delays (already discussed in section 3.4).

The final Roadmap is similar to the initial one having just a few modifications regarding test-harness vs simulation, MQTT broker vs back-end of the UI, agents vs devices abstraction, external problems (slow network, signal loss, hardware issues) vs diversified scenarios (with different input variables -

number of calls, number of devices, call duration and delays, profiles).

This capstone project is iterative in many layers and therefore the roadmap is also iterative which means that after completing the last step it should begin (almost) from the top.

The **initial** proposed roadmap had the following procedure:

1. Familiarization with the Presence API and understand what are the request type and the specification of the workload we would use to test the system.
2. Create a simple back-end that would make the connection to the MQTT broker and further in we would had concurrency to the play.
3. Add mock Callbar agents. This agents would bring to our system several scenarios with their status and their behaviour.
4. Create test cases for possible failures.
5. Retrieve relevant metrics from the test cases.
6. Analyze metrics.
7. Present the metrics in a UI.
8. Propose solutions (if any).

The **final** roadmap we made had the following steps:

1. Familiarization with Talkdesk's APIs. Especially the RP API and this module functionality – which had little documentation because it was brand new.
2. Creation of back-end communication with the RP through API's endpoints. Make HTTP requests (and test it for multiple requests - *GET*, *POST*, *PATCH*, etc) and analyse concurrency.
3. Create and add mocked Callbar agents beneath the representation of device on the RP. Test scalability on the back-end side with the injection of small to large quantities of device list requests using the RP.
4. Develop a first simple base version of the front-end of the interface with the main visual blocks and interaction with the user by inserting some input values.
5. Create test cases and different scenarios, personas and behaviours – listing the default ones in the simulator that may be used and selected and also allow the user to custom his own scenario with the input variables he decided.

6. Integrate back-end and front-end following the system architecture (section 4.4) and allows the interchanging of data and information on setup and whenever an event occurs.
7. Run the simulation and watch the real-time responses on the interface – timeline graphic, devices' activity and filtering, simulation test running display.
8. Gather useful metrics – time elapsed, number of successful accepted calls.
9. Analyse and produce reports with the gathered data from the simulation.
10. Present the DEMO, receive feedback and propose solutions (if possible).
11. Iterate to step 3.

4

Callbar Simulator

Contents

4.1	Business Model Canvas	45
4.2	Development Environment	48
4.3	Project Phases	48
4.4	Software Architecture	52
4.5	Back-end	54
4.6	Integrations	56
4.7	Front-end	58
4.8	Restrictions and Limitations	64

This Chapter describes our product – Callbar Simulator – and has more detailed information about the development phase of the capstone project. It includes the project development phases, each phase’s main tasks, the BMC of the project and some other important references.

4.1 Business Model Canvas

BMC is a visual strategic management model template that is useful to understand a business (or simply a product) development and its elements described in a visual chart. With this model we can see the Callbar Simulator infrastructure, its value propositions, the customer segment and also the finance value of the product.

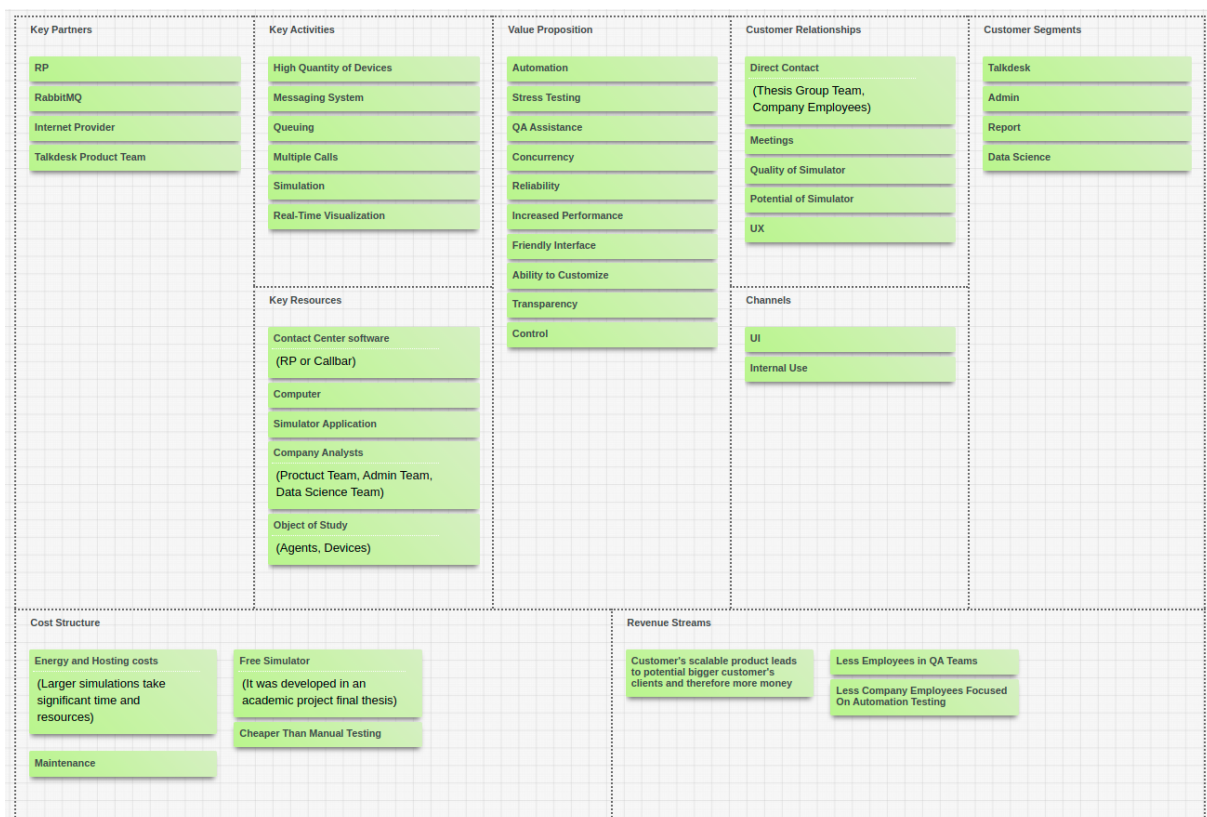


Figure 4.1: Business Model Canvas.

As may be observed in Figure 4.1 Callbar Simulator has the following sectors:

Customer Segments

Callbar Simulator serves its customer target which is Talkdesk – especially useful for Admin, Report and Data Science teams that may evaluate and analyse the results of the simulation.

Value Proposition

The values we deliver to our customer segments are in accordance to the customer (Talkdesk) needs (section 2.1.4) and they are what differentiates our simulator from already existing smaller processes that the company uses. The main value propositions are:

Automation - After setting up the simulation it should run automatically until the completion criteria.

Stress Testing - The user decides how much load he wants to perform on the product test.

Quality Assurance (QA) Assistance - These teams are usually smaller than the company needs and have too much work on their hands and a complete simulation testing a product possible for hours in a row may be very dull.

Concurrency - The simulation performs many concurrent requests and many modules work simultaneously.

Reliability - With the simulation it is possible to observe in real-time if the scenario is reliable and if could satisfy a real client expectation.

Increased Performance - The Simulator allows the customer to get the critical point of product usage and work on the critical path and possible worst case scenario to increase the performance of the product.

Friendly Interface - A user interface easily manipulated by the customer's employees being them tech or non-tech.

Ability to Customize - At the setup stage of the simulation the user may customize his test scenario, his profiles and the features of the simulation.

Transparency - The simulation allows the user to observe what is happening in real-time – tasks sometimes hidden that are just observable in the back-end of the operation.

Control - When a possible issue occurs, the user is able to control its original source and mitigate the problem. Product control is close related to transparency and is very useful when increasing the dimension of the simulation and scale of the product.

Channels

Channels refer to how the simulator is delivered to Talkdesk and this is simply done by internal direct delivery for the company to use by their teams. They just need to receive the UI and its source code.

Customer Relationships

Callbar Simulator is considered a product that was developed somehow internally and therefore the customer relationship includes direct contact, meetings and is maintained by the quality and potential growth of the application and also from UX.

Revenue Streams

This simulator does not bring customer's direct profit but allows to increase its value and profit in the future mainly by scaling the customer's product to bigger clients and therefore get more money from them. It also allows Talkdesk to save money on employees directly working in QA teams allowing these people to perform other kinds of testing and it helps on the automation testing.

Key Resources

Key resources describe our most important strategic assets to make the model work. This includes potential physical, human, intellectual and financial. Therefore Callbar Simulator needs a contact center software (might be Callbar, RP or other component), a computer to run the simulation, the simulator application, human or computational Talkdesk analysts (could be from the product team, admin team, data science team, etc) and the objects of study (agents, devices, calls, etc).

Key Activities

These are the strategic things that must be performed to make the model work and are directly relatable to the value proposition. Key Activities include simulation, high quantity of devices and calls, real-time visualization and control, message system and queuing.

Key Partners

In this block we describe the partners needed to perform the simulation, besides the direct assets already described. The key partners are the Callbar RP, RabbitMQ, internet provider and Talkdesk product team.

Cost Structure

In this sector it is mapped the key activities to the inherent costs. One main advantage of Callbar Simulator is that it has very little cost structure being it just energy and hosting costs because larger simulations take significant time and resources to be performed although cheaper than manual testing, maintenance of the simulator software and this is a free simulator as it was developed in an academic project final thesis.

4.2 Development Environment

The environment for this project development was divided into physical and digital domains. This environment also included all the entities described in section 1.3.

The physical one is where the team was (or some team member) developing the project and the main local was the team table at the open space of Lisbon's office that Talkdesk kindly provided for the project development. This space was available to us whenever we would like to be there but due to our availability and individual schedules we compromised to be there at least the first three days of the week from 2PM to 6PM simultaneously. Besides this work schedule we often met at Talkdesk facilities with company employees and teams. Outside of the company office I developed my portion of this project at home, IST (both developing and meeting with the team, assistant professors and another capstone team) and also at FBA with the rest of the team in the final stage of the work for the website, video and presentation preparation and elaboration.

The digital domain includes obviously our own personal computers, the project hosted location and all its digital and source components locations, help tools and communication technologies. For communication we used the company *Slack* group, our own team's *WhatsApp* group and email. We shared digital resources with email, *Google Drive* and *Keybase*. The project was hosted in *Github* and almost at the company's servers and the other materials (videos, presentation, website) were hosted at the team's *Google Drive* directory and also at personal domain of team members IST [website](#). For our Agile work methodology we used Kanban board of *Jira* software after we had previously used *Trello* for that matter.

4.3 Project Phases

In this section I will state the stages of the development of the final project and not the ones already discussed in section 3.4 relative to an initial scope.

This capstone project had the following stages (from the first to the last ones) also described in Figure 4.2.

4.3.1 User Research

User Research (UR) helped us understand the project potential, the Callbar and RP usage and define different scenarios, personas and profiles based in the shadowing, interviews and meetings with the company teams. UR is described in section 3.2.

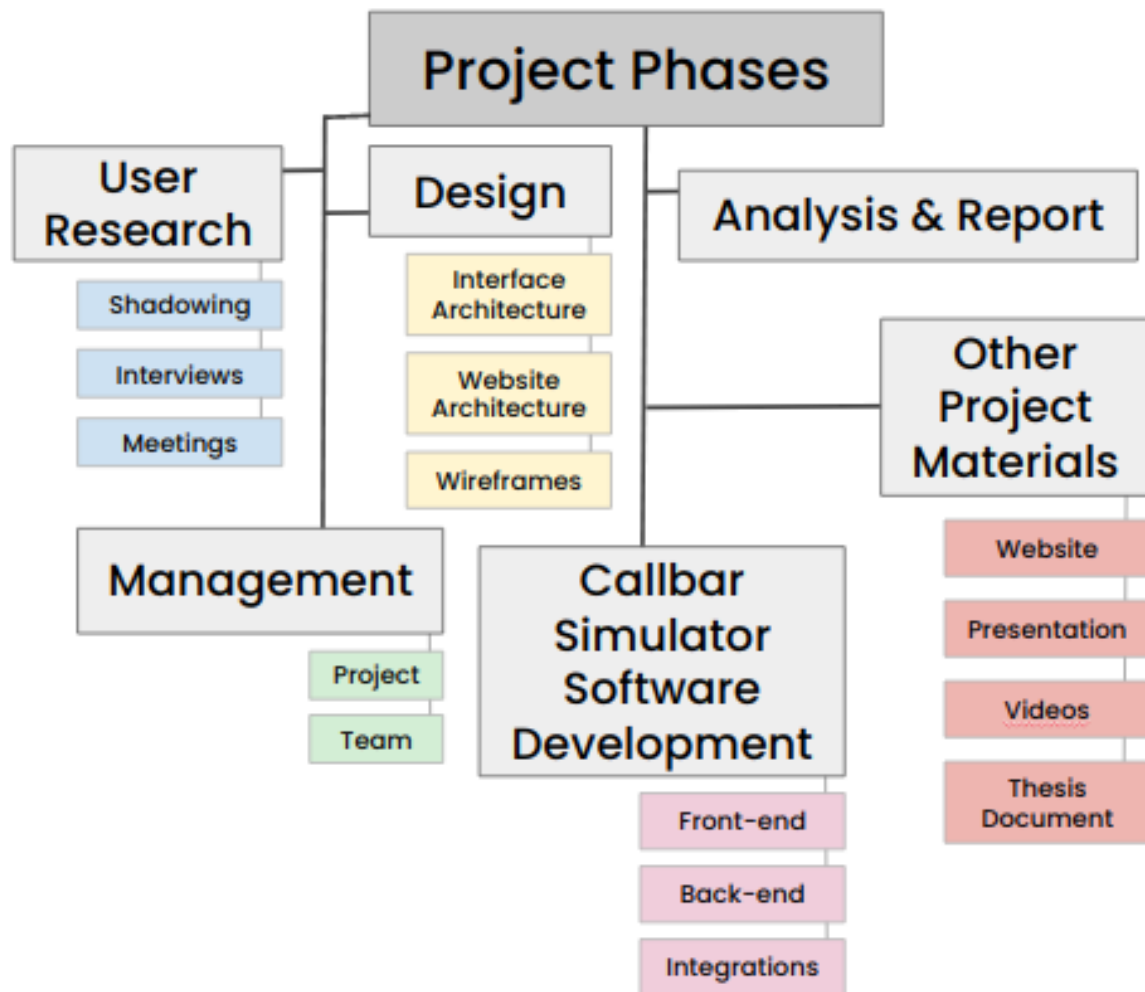


Figure 4.2: Project Phases.

4.3.1.A Shadowing

We performed *shadowing* (3.2.2) on Talkdesk's support team and also on sales team because they used Callbar in their work day although with some limitations as the support team used other tools and services too and just used Callbar occasionally. Sales team used Callbar to call potential clients trying to sell Talkdesk services but almost only used Callbar to make outbound calls and our simulation only involved inbound calls so we gathered information about users and their behaviour when they wanted to connect to other agents but we did not quite gather user information and their behaviour when answering a ringing call.

I personally performed Shadowing to two people from Sales Team – An example may be seen in appendix A.

4.3.1.B Interviews

We made some interviews (see section 3.2.3) also to support and sales team which used Callbar. Interviews were made after the Shadowing and were richer in terms of user characteristics and input ideas to the project.

I had several informal interviews where the questions of appendix B where asked and answered.

4.3.1.C Meetings

Meetings were regular at IST and Talkdesk. At IST we met with assistant professors informing them about the situation report and also receiving input about the capstone project. At Talkdesk we met with many teams (as discussed in section 3.2.1) and we also had daily team meetings, a planning meeting and a retrospective one because of the Kanban Agile methodology we followed (section 3.5).

4.3.2 Team and Project Management

This phase was relevant from the beginning to the end of the project and although supervised by our team member project manager, all members contributed to this phase as it initially ran on *Trello* and then on *Jira* software usually with task cards which each team member edited and managed in team work.

The Agile methodology we followed was very useful and oriented to this phase.

4.3.3 Design

Design was the major contribute of our UX/UI designer colleague that performed most of the work regarding the tasks described below. My input to this phase was a creative one giving ideas and opinions about the tasks and components and contribute to the team brainstorming.

4.3.3.A Interface Architecture

Interface architecture is the model that guided the team when developing the front-end of the application (Callbar Simulator). It includes the screens, the general information present at each one and the flow between screens.

4.3.3.B Team Website Architecture

Same as the previous task but for the informative team website ([at our site](#)).

4.3.3.C Wireframes

These are basically initial prototypes that could be drawn in paper or image editing. They present information that should be displayed in the interface, gives an outline of the structure, layout and style, carries an overall description and direction of the interface and it is the bridge between paper sketches and the first prototype that should be a common language to designers, users and development team (especially to front-end – section 4.7).

4.3.4 Callbar Simulator Software Development

This is the core phase of the project as it contains its content and development. All the previous phases and tasks described are very important to this phase and this is an iterative work in progress until the completion of the project.

This development was mostly performed by the two (me and my colleague) software engineer team members being my contribution towards the front-end and divided towards the integration front-end back-end.

4.3.4.A Front-end

Further described in section 4.7.

Front-end must follow the guidelines provided by the design phase and be as authentic and similar to the wireframes as possible.

In this phase it was developed the various screens and visualization blocks and their functionalities and features – Graphic Timeline, Device Activity, Running Simulation and Scenario Setup.

4.3.4.B Back-end

Further described in section 4.5.

Back-end must perform the simulation and interact with the company's API assuring security, quality, queuing and parallelism. This phase is not visible to the end user of the simulator but is responsible for the management of data and information and includes core operations like presence system, call assignment, variability and machine state transitions.

4.3.4.C Integrations

Further described in section 4.6.

These integrations refer to the protocol and data stream from back-end to front-end, and from front-end to back-end too. The technology used was *socket.IO* as it listened and transferred files to deter-

mined endpoints whenever an event occurs in real-time. So from front-end it was streamed the setup scenario in the beginning of simulation and from back-end it was streamed files with up-to-date information every time an event occurred or a state transition happened.

4.3.5 Analysis and Report

At this stage it was possible to study the simulation output visualizing it on the test view block screen. It had useful information regarding time elapsed and number of devices and calls answered.

As may be seen in the next chapter section 5.2 for future work it may be developed a report feature in the visualized information persistent in some file to be downloaded and later be manipulated and analysed accordingly.

4.3.6 Other Project Materials

Because this project is a capstone one and is included in an academic thesis, some material besides the Callbar Simulator itself must be developed – an informative website with the product (Simulator) description, team personal area and all relevant content about the project; a team presentation for the professors with the practical project and demonstration (which already existed); an individual presentation with each member contribution and personal analysis for the thesis presentation; product videos with information and commercial-like.

4.4 Software Architecture

The Callbar Simulator has a few layers of abstraction and may simply be described having a front-end application – user interface – that is web based in the browser and it just communicates with the human end user and the back-end of the application through a two-way web socket – *socket.IO* – with endpoints to localhost. The back-end is connected with Talkdesk's API of RP, uses *queuing* and messaging system with *RabbitMQ* over *AMQP* protocol.

Some files are accessed locally present in our machines and are just access files, credentials and also *JSON* files we created for the front-end to load default scenarios and profiles.

My main contribution to this project is resumed in the front-end architecture that may be seen in Figure 4.3 and there is a bi-directional communication between front-end and back-end in webSockets using *socket.IO* in which the front-end sends a *JSON* file to the back-end simulator (at the beginning of each simulation) with the scenario's variables the user desired to simulate and the back-end sends to the front-end application a *JSON* file with the updated list of devices (with their current states) every

time a change occurred in the simulation (in real-time). At every second the application updated two components – time elapsed since the start of the simulation and the graphic timeline (with *D3.js*). The end user may interact with the Callbar Simulator by setting up the scenario to run and then by clicking in the features of "Stop" (or "Reset") the simulation, filter device activity list by some status label and changing the width and range of the time dimension in the graphic timeline.

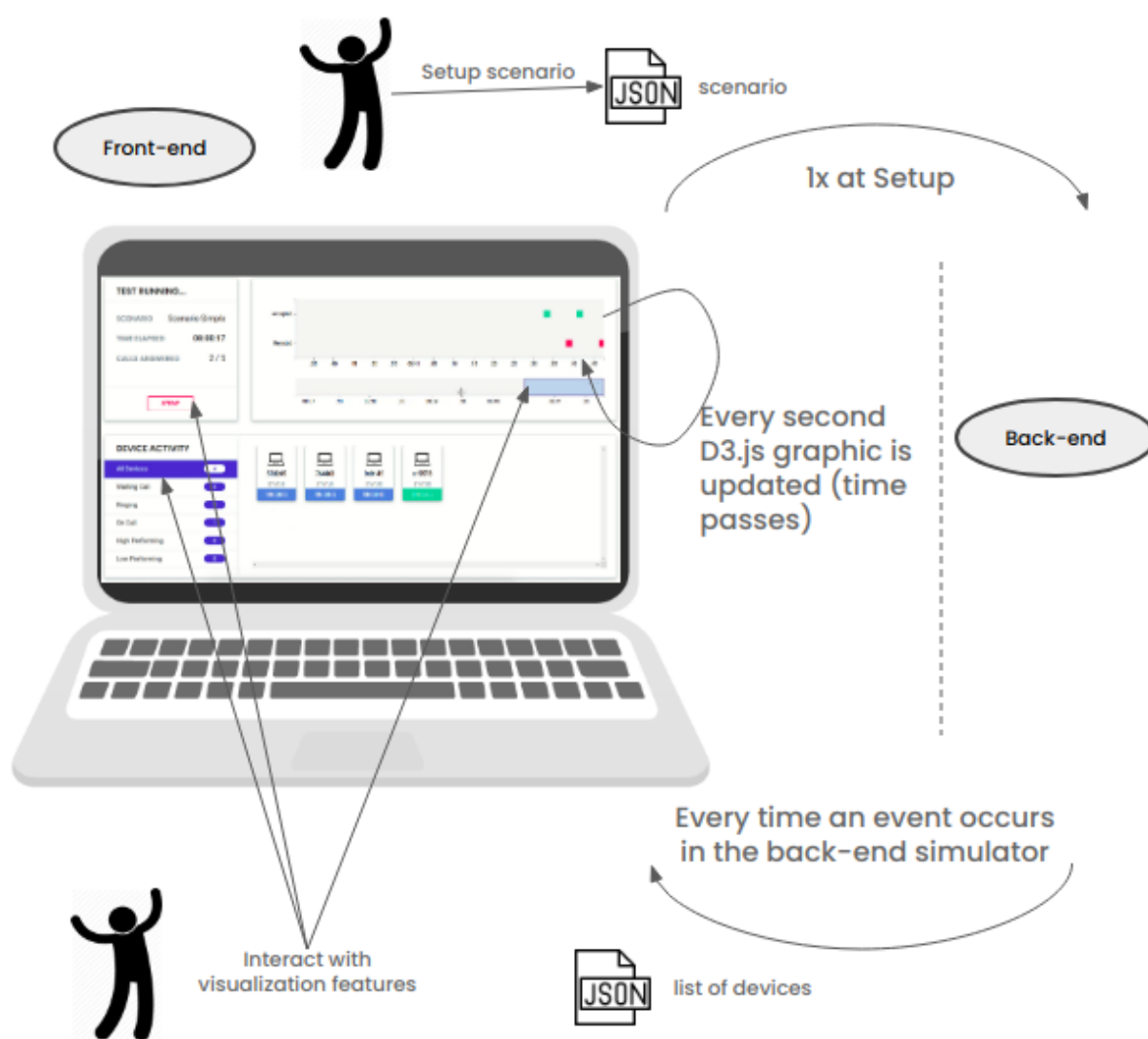


Figure 4.3: Front-end architecture.

4.5 Back-end

Back-end is the data access and manipulation layer of a software project. In our project the back-end is considered the brain of the operation and is responsible for providing the correct functionalities to Callbar Simulator and interacting with almost every component and external modules present in the software architecture (section 4.4) including Talkdesk's API, messaging system, queuing, parallelism and should stream correct information to the front-end of the application. This layer is between the hardware components and the front-end visualization and handles the system logic and data manipulation – which is hidden from the end user of the simulator.

4.5.1 Pseudo Presence System

This initial version of talkdesk's product (RP) still had not the presence system developed so we needed to make our own simple presence system following the following simple Finite State Machine (FSM) (Figure 4.4) which was developed in group and in accordance with Talkdesk's product concept.

This FSM includes the possible states of each device for a call. A device is an abstraction of an agent. There are three possible states: **Waiting Call** (same as "Idle"), **Ringling** (same as "Pending Call") and **On Call**. At the beginning of the simulation it is created the specified number of devices and all begin with their state "Waiting Call". From this state it is only possible to go to the "Ringling" state when an inbound call appears and at this moment all devices that were waiting for a call must go to their ringling state. From the "Ringling" state a device could go back again to its initial state waiting for a call if the call was rejected or go to the "On Call" state if the call was accepted by that device. When a device has its state "On Call" it should remain there for the time period specified in the call delay at the setup stage of the simulation and after this time is up the transition of the state "On Call" must occur to the initial "Waiting Call" state.

4.5.2 Assignment

As there is no information about clients and users (meaning that the RP has no information about the device or agent that should get a specific call for a customer) we had to develop a simple assignment system and choose some device to have the initiative of picking up (or rejecting) some call because this is an automated simulation.

So when a call is ringling it should ring to every device on the "Waiting Call" state (ringling group) and then we randomly choose a first device candidate of the "Ringling"/"Pending" state to accept or reject the call. If that device rejects the call, we randomly choose another device from that state list and do the same. If the device accepts the call then all the remaining devices reject it and go to the initial state.

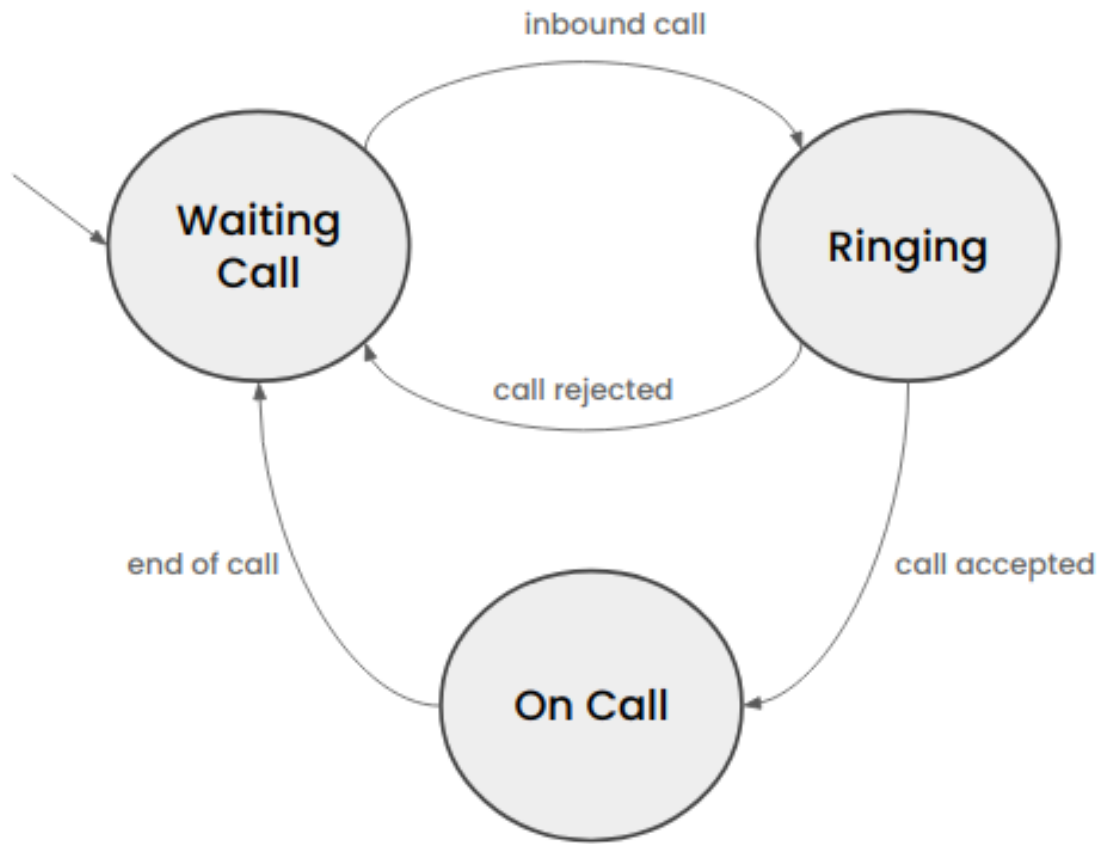


Figure 4.4: Finite State Machine for presence system.

The decision of a device to accept or reject the call is done in real time by the device's call pickup probability (inherent from the profile defined in the setup stage of the simulation).

4.5.3 Variability and Profiles

Each profile has a probability of picking up a call (0 to 1) which contributes to variability, randomness, entropy and complexity of the simulation. This device pickup call probability is an attribute of each device deriving from its profile.

Each profile represents a persona (described in section 3.2.6) and we created four different personas after the user research previously done. The profile name is descriptive but it links to a defined float value (0 to 1).

Profile 1 - *Shy* - It was considered that an agent (device) had a probability of 0.1 (10%) of answering a call.

Profile 2 - *Champion* - It was considered that an agent (device) had a probability of 0.9 (90%) of

answering a call.

Profile 3 - Normal - It was considered that an agent (device) had a probability of answering a call resulting from the computation of a normal distribution function.

Profile 4 - Sleepy - It was considered that an agent (device) had a probability of answering a call resulting from the computation of an exponential distribution function.

4.6 Integrations

Data should flow between the presentation layer (front-end) and the data access layer (back-end). This interchanging of data was performed using webSockets with *socket.IO*.

The data format we choose to flow were *JSON* files because they are easily manipulated both by Python in back-end and JavaScript in front-end and it's a technology very used nowadays in tech companies.

The front-end of the application had to integrate the stream of information with back-end and the usage of *d3.js* as a JavaScript library for the real-time interactive timeline graphic.

The back-end of Callbar Simulator had more integrations such as also the communication with front-end already discussed, communication with Talkdesk's API of RP, messaging system trough *RabbitMQ* with *AMQP* protocol and other external modules usage. Therefore the back-end was the central processing unit of our simulator whereas the front-end was the connection between the simulation and the end user.

4.6.1 Talkdesk API

We connected to Talkdesk's API firstly to get the authorization and authentication and then to receive events from some modules (testing and using RP) – for this matter we used endpoints and secret credentials from Talkdesk team.

To get information from API we just needed to make a request (HTTP request GET, POST, PATCH) with the correct header and then receive a message response – usually a *JSON* file with the desired content.

An example of a Talkdesk API sample request might be seen in appendix C.

4.6.2 Protocols and Data Structures

For the development of the project we had to respect some protocols of data transmission, security, internet, messaging system and data structures.

WebSocket is a two-way communication protocol providing stream channels over a TCP connection. It is compatible with HTTP protocol – WebSocket handshakes with HTTP to change from one protocol to another. This protocol was very useful to us because it enabled interaction between a web browser (our front-end application) and a server (our simulator back-end) with lower overhead assisting our real-time events (data files) from and to the back-end/front-end.

socket.IO was the technology we used to follow the WebSocket protocol. It is a JavaScript library for real-time events and therefore it is event-driven. This framework is bi-directional between client and server. Although socket.IO is a custom real-time transport protocol it is needed in both client-side and server-side to be implemented using socket.IO.

HTTP is the foundation of data communication for the *world wide web* and is in the application layer. It works as a request-response protocol in the client-server paradigm. HTTP has some methods in the request message and the main ones (that we used) are: *GET* and *POST*. The first one only retrieves data and has no other effect. The second one might be a data retrieve request or/and some update for the server.

TCP is an internet protocol suite in the transport layer and its key features are: ordered data transfer, re-transmission of lost packets, error-free data transfer, flow and congestion control.

AMQP is an application layer protocol for message-oriented middleware which is commonly used in back-end processes not visible by most of the application end users. The key features of this protocol are: message-oriented, queuing, routing, publish-subscribe model, point-to-point model, reliability and security.

JSON is human-readable text that may be understood both by tech and non-tech people. It follows a simple syntax schema and its data types are: string, boolean, array, object, number (integer or decimal) and null. It consists in attribute-value pairs and is very used in many modern programming languages although deriving from JavaScript.

Our data format in *JSON* files was very simple and object oriented with previously defined (by the team) names for objects and their attributes. An example of our data in a *JSON* file is in the appendix D.

4.7 Front-end

Front-end is the presentation layer of the application – usually deployed in the client side – and the direct connection between the user interface (simulator) and the user, providing a user-friendly interface focused in accessibility and performance. It is the responsibility of the front-end to enable interaction, visualization, creativity and user experience of the application. It makes the bridge between the team members of design and back-end and therefore their contributions and development.

It receives input from the user at the beginning setup of the simulation being it just the selection of a predefined scenario, profile and environment or the customization of the scenario by the user.

While the simulation is running the user may observe in real-time what is happening and interact through a list of devices with their activity (that may be filtered by device status) and a graphic timeline flowing in time.

My main contribute to this project is described in this section – front-end development.

The front-end development of the Callbar Simulator may be divided into four subsets:

Loading Page - Just an initial loading page before the real beginning of the application (see figure 4.5).

Initial Screen - Just a simple and clean visualization of the main features and a clickable block for the user to setup the scenario to be simulated (see figure 4.6).

Setup - In the "Setup" step (prior to the simulation beginning) the user needed to click on the "Run" button to start the simulation. If no other action was performed by the user, then a default scenario is loaded from a JSON file list of predefined scenarios. Instead of just running the simulation, the user may see the details and select a predefined scenario from a list of scenarios (loaded from an external JSON file that may be created by the user, or not – see figure 4.7) or customize his own scenario by filling the variables input boxes and selecting the desired persona profiles (see figure 4.8). In any case the output of this simulation setup phase is a JSON file to be sent to back-end with the variables:

- Number of devices
- Number of calls
- Duration of each call
- Delay for each device to pickup the call

- Each call ringing time (standby)
- Four persona profiles (may be repeated) - "Champion", "Shy", "Sleepy", "Normal".

Simulation - This is the main subset of the front-end of the project and has both **automation** and **user interaction**.

- **Automation** - At a specified timeout (every second) of the simulation the Hyper Text Markup Language (HTML) components (Document Object Model (DOM) elements) time elapsed and graphic timeline (with *D3.js* library) are updated both because time passes and therefore the HTML page is rendered even if no (other) alteration happened in the simulator.

The other way that forces the front-end application to update is by receiving an event from the back-end simulator with an updated list of devices and then it updates the device activity component, the devices lists by label and status and also the visualization of a call accepted or rejected (if applicable) in the real-time graphic.

- **User Interaction** - The user may interact with the simulation application in two aspects: either by setting up the scenario or clicking in the visualizing components during the simulation. To setup the simulation the user either chooses a predefined scenario (see figure 4.7) or customize his own (see figure 4.8).

At any time during the simulation, the user may click in some sections and interact with the application (also discussed in section 4.7.3):

- **"Stop" button** - Stops the simulation and goes back to the setup step (in order to begin a new simulation) – figure 4.10.
- **"Reset" button** - Same as the previous button but in the case the simulation had already finished.
- **Graphic Slider** - Change the width and range of the timeline graphic.
- **Device activity labels** - Filter the list of devices by their states ("All Devices", "Waiting Call", "Ringing", "On Call", "High Performing", "Low Performing") by clicking in the respective label.

4.7.1 Visualization

In this project we refer to visualization as data visualization. It is a graphic representation of information to be observed by humans.

This data information is shown in the application user interface in the browser and is represented by texts, images, videos, visual blocks, charts, graphics and animations. Effective visualization enables

users to better understand the simulation and increases user experience. Our main goal in the application front-end was to make a user interface as simple as possible, only with the relevant information for the user to see and to interact (see section 4.7.4 with the application screens examples) and still performing the desired feature functionalities.

Data visualization is an art (therefore the design team member input) and also a science (my personal input and contribution). Front-end follows the design work and wireframes previously made in image editors or in plain paper (initially) from the design phase.

Callbar Simulator is a web interface that runs in the browser (it was tested in *Chrome* as it is also the browser chosen by Talkdesk to make support of its products) and we focused our development for the desktop version because it doesn't make sense to perform a product simulation in mobile where the visualization screen is much smaller and computational power of the device is shorter too. Nevertheless Callbar Simulator also works in mobile with the desktop view although with much smaller visualization blocks.

Visual perception of data is different among the living beings and since our project was made for Talkdesk teams we were only interested in the human perception. Humans can distinguish differences in colors, orientations, shapes and text fonts (among others) and the style choices may significantly increase users familiarity and control of the interface reducing time and effort to perform their desired tasks. Human cognition are people processes like memory, learning, attention, problem-solving, intelligence and perception.

When designing and developing the visualization aspects of the interface, the team chose to give more importance to intuitive visual content because human visual processing is efficient in detecting changes in dimensions, quantities, colors, shapes and animations.

D3.js - Graphic Timeline

D3.js is a JavaScript library for generating dynamic and interactive data visualizations in web browsers and it allows great control and transparency over the visual results. The technologies used in *D3.js* are HTML, Cascading Style Sheets (CSS) and Scalable Vector Graphics (SVG).

4.7.2 Style

The Callbar Simulator has its own style, defined in the front-end code, represented in *CSS* files. This overall style was created in the design phases and I tried to replicate it in the interface code.

The application style contributed a lot towards the visual aspect of the simulator.

4.7.3 Interaction

When the user first opens the Callbar Simulator, a static loading screen appears for a starting approach of the application and to warm up the user. This takes about five seconds and some calls so back-end and then to various APIs may be performed before the simulation really begins.

Then the user needs to specify or choose the simulation environment interacting with the setup box being it predefined scenarios or a custom one. This interaction occurs just by clicking at elements and filling some input boxes the user desires, to customize the simulation with his own delays, times, and number of objects (devices, calls) variable values.

While the simulation is running the user is able to observe real-time events (calls being accepted and rejected) and the activity of its devices and their current state ("On Call", "Ringing", "Waiting Call") but the user is also able to interact with the application customizing the timeline graphic on its time window width and size, and filtering the device activity block by the desired states of the devices.

Also related to human interaction with visualization systems and interfaces are visual analytics. They are part of the science that studies analytical reasoning supported by the interactive visual interface. The goal is to deal with massive information and dynamically changing information spaces and research perceptual and cognitive operations – our project had a little bit of this topic although we abstracted to the point of releasing all the information regarding the visualization of all devices with a scroll option and a filtering feature, and for the real-time events graphic we used *D3.js* library.

4.7.4 Screens

Objectively the interface is a web interface with only two screens – loading screen which is irrelevant for the simulation operations and the Callbar Simulator screen. This last one has various components and some of them are hidden when others are displayed making the illusion of screen changing for the end user. Besides this adopted approach I consider each visualization block to be a screen as it contains specific information about some aspects and the user attention flows from one block to another and therefore I consider it to be the flowing between screens.

The simulator screens are described and illustrated bellow:

Loading Screen - It is just a simple static screen with the simulator name and a loading animation as we can see in Figure 4.5 – it waits five seconds and then changes to the simulator screen.

Simulator First Screen - It has three blocks – "Run Test", "Call Timeline" and "Device Activity" as we may see in Figure 4.6 – where it only possible to select the first one. The other two have just a

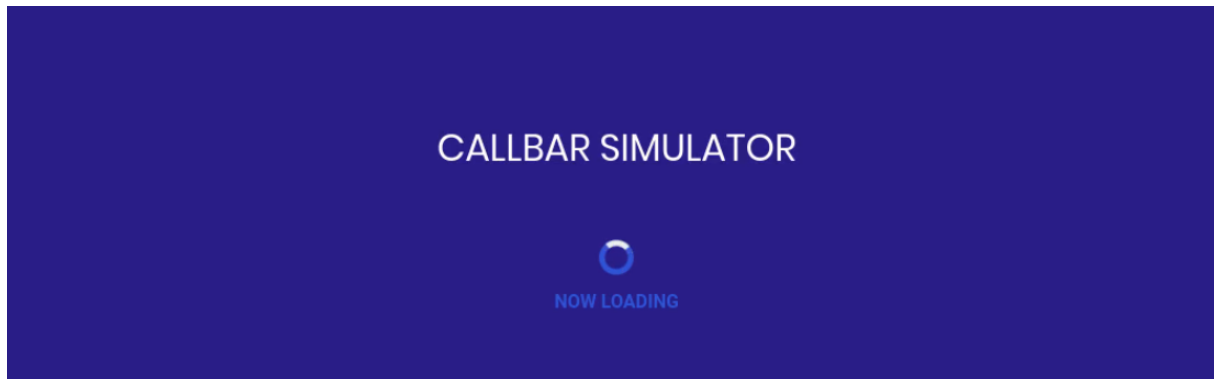


Figure 4.5: Loading Screen.

description of what will happen there.

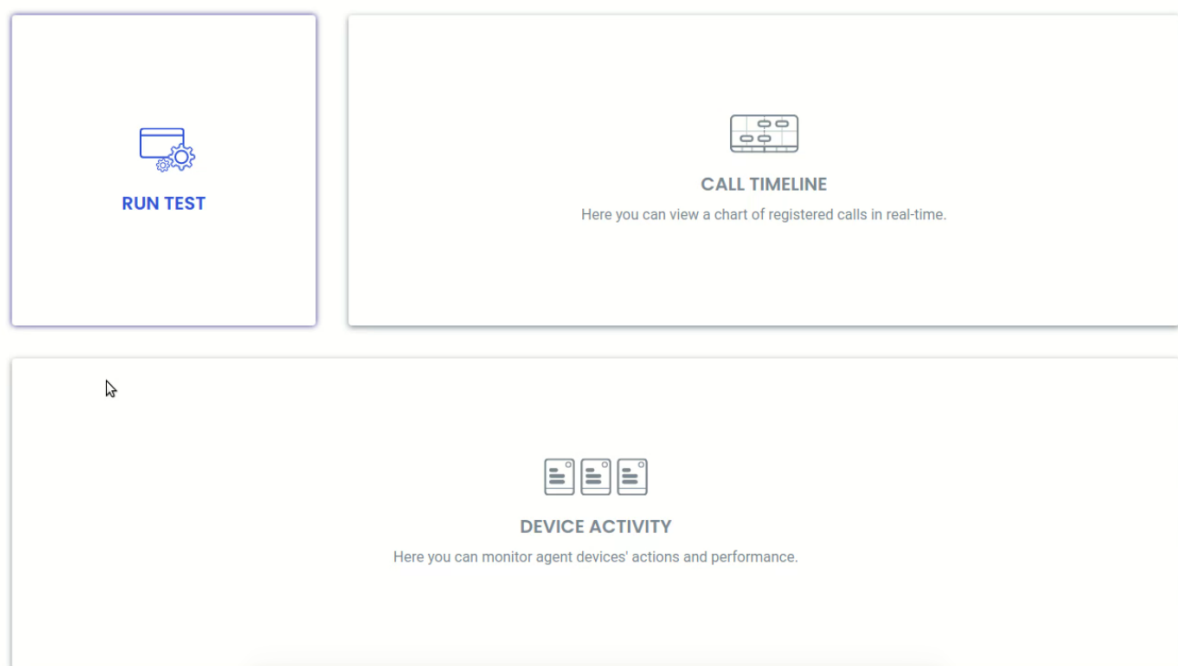


Figure 4.6: Simulator First Screen.

Test Setup Screen - After clicking in the "Run Test" block of the previous screen the user is prompted up with the "Test Setup Screen" with the presets scenarios list to interact. Here the user may click on "Run" the simulation after doing nothing (in which case a pre-defined default scenario is loaded) or select a scenario and see its details (Figure 4.7). It is also possible to customize the scenario by clicking "Custom" and filling the input boxes with the desired values for the simulation and selecting four device profiles from a list – "Champion", "Shy", "Sleepy", "Normal" – (Figure 4.8).

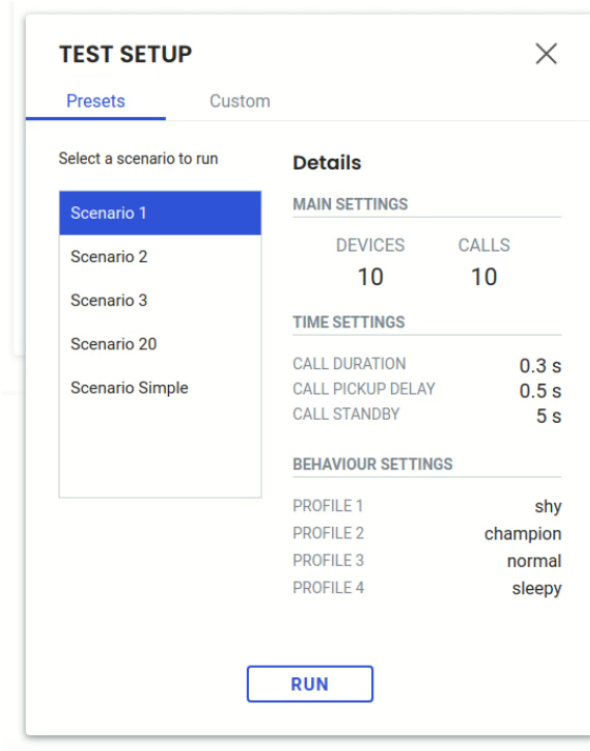


Figure 4.7: Test Setup Preset Scenarios.

Simulation Screen - After clicking "Run" on the previous screen, the simulation begins. In this main screen there are three visualization blocks – "Test Running", "Device Activity" and the timeline graphic – as seen in Figure 4.9.

The first one has general information of the simulation – selected scenario name, time elapsed since the beginning of the simulation, number of answered and total calls and a "Stop" button as well.

The second block has the list of all devices with their identifiers and status being updated in real-time. This feature allows to scroll if there are many devices and to filter the device list by category – "All Devices", "Waiting Call", "Ringing", "On Call", "High Performing" and "Low Performing". The third block refers to the real-time graphic with the events occurring – when a device accepts a call or rejects it. This block is interactive and the user may modify the time window width and click on the graphic.

The user may click on the "Stop" button or "Reset" button (if the simulation have finished) to return to the "Simulation First Screen".

TEST SETUP

Presets

Custom

MAIN SETTINGS

Number of Devices

10

Number of Calls

10

TIME SETTINGS

Call Duration (ms)

300

Call Pickup Delay (ms)

500

Call Standby (ms)

5000

BEHAVIOUR SETTINGS

Profile1

Shy

Profile2

Champion

Profile3

Normal

Profile4

Sleepy

RUN

Figure 4.8: Test Setup Custom Scenario.

4.8 Restrictions and Limitations

My personal opinion about this project and the issues that appeared in our way will be presented in the conclusions section 5.

There were some limitations and constraints to the project development that we faced and had to deal. First of all this whole capstone project is an innovation at my university in the final thesis of my course so there was no other similar project to compare or guide us – the paradigm has changed and with it some particularities regarding the traditional course evaluation. We suffered an important limitation in terms of time management when we needed to wait several weeks for the company product new version and then it wasn't ready and deployed yet so we add to deal with a different product – a core module RP new version – and modify the project initial scope adapting our previous user research and previous project bases already prepared. Instead of a test-harness we changed to a simulator. Resuming we had a limitation that was the dependency of other teams, in this case the Talkdesk product team.

Also about the RP, this new product had very little documentation since it had just been developed.

Another restriction was a tech one regarding the communication between front-end and back-end. We needed to interchange data files (*JSON* files) two-ways and in real-time whenever an event might

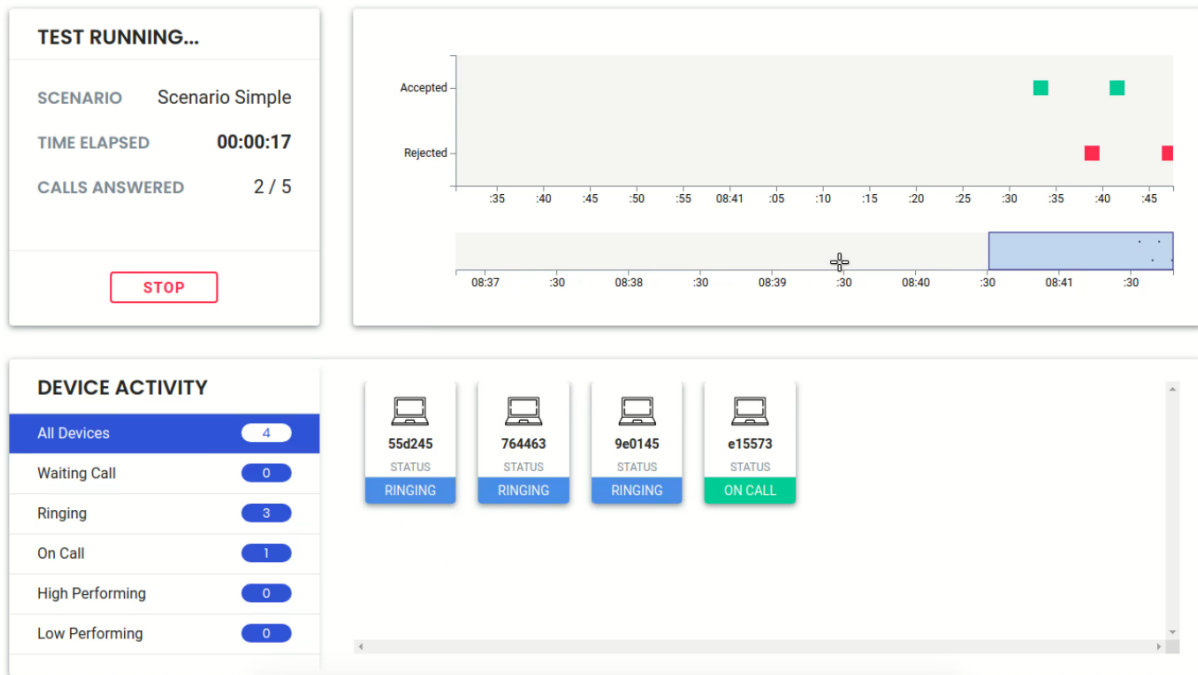


Figure 4.9: Callbar Simulator Main Screen.

occur in the back-end coming from the company's APIs or simply from our simulation computation – for this problem we decided to use web sockets with *socket.IO* technology.

We also had a physical local limitation when we lose our space at the company's open space in the final stage of the project due to the company's personnel expansion and therefore space occupation so the team adapted and worked at other spaces both still at the company's facilities and also outside of it.

Regarding team and project management we had some difficulties learning how to use *JIRA* software since it is a complex tool and none of the team members had experience in that matter.

TEST RUNNING...

SCENARIO	Scenario Simple
TIME ELAPSED	00:00:15
CALLS ANSWERED	2 / 5

STOP

Figure 4.10: Test Running Information.

5

Conclusion

Contents

5.1 Results	70
5.2 Future Work	70

This capstone project had the duration of a year (two semesters) as it was included in both my Master Project and Master Thesis in Information and Software Engineering. The first one was on the first semester and corresponded to the research of the project while the second one corresponded to the second semester work on the project development.

The team project result was a product – Callbar Simulator – to be used by the target company Talkdesk. The project suffered some modification in its scope and changing from an automated test-harness to an automated simulator. Nevertheless the project output is very similar as it may test massively Talkdesk's product being it Callbar or a simpler module component – RP – and stressing it to the critical point so Talkdesk can analyse the issues that occurred and then act accordingly in order to increase the scale of its product always testing it and making sure it is reliable.

Project research and development information towards the thesis are described in this document. In this report I intended to describe the progress of the project from its initial research to the final presentations and its various components. Although this was a capstone group project, in this document is my contribution more detailed, the team work developed, concepts described and research performed as well. Also this document thesis has my personal view of the thematic, project analysis and opinions.

This project was a tech demanding one because it required software engineering and development knowledge and a large part of my personal research was directed to this technological path. Not only I had to increase my software development skills I also had to understand the big and complex software architecture of the company and the connection between the various modules Talkdesk has in their products and then understand what is required and returned of the component RP of the new version that did not have relevant documentation because the product development teams were still developing and finishing that same version.

With this capstone project I personally acquired many skills and improved existing ones that I already had. More specifically I increased my technological stack in terms of programming languages such as Python and its libraries, JavaScript and its libraries, JSON, HTML and CSS. The constant communication and the development methodology adopted – Agile, Kanban – also enabled new knowledge and paradigms to be acquired, for example team work, regular meetings, briefings and debriefings, presentations and networking between the team, the assistant professors and the company employees and their teams too. The connection between the last phase of the academic course (final thesis) and the very first step in the work environment was demanding and interesting because issues came from everywhere the it was our responsibility to overcome them in group using (or creating) our problem-solving skills.

It was possible with this project for me to learn about my field of study (software engineering) and also contents of design, team and project management, user research, communication, critical thinking,

problem-solving and other soft skills.

5.1 Results

The practical part of this project was presented in June of 2019 to both the professors and the company while the company's product stable version was available to us and a live demonstration of the product was performed – a video demonstrating the Callbar Simulator may be seen at [site.com] and the promotional videos at [our site](#) which is full of information about this project.

Some of the research results may be analysed in chapter 3 and some of the development results may also be observed in the chapter 4 of this document.

Our main goals with this project were fulfilled as the simulator allows Talkdesk to perform automated testing and therefore test its product – not only RP but it is also extensible to Callbar or other module components as our project is modular and uses the return messages of Talkdesk's API as a product so in order to change the product it is just needed to change the target API. We consider the results positive because Talkdesk representatives (that were our main point of contact inside the company) were happy and gave good reviews when the Callbar Simulator was presented. Our product has potential to grow and increase value on a target company product or service and on top of this it is simple but has quality and gives a good user experience when used, increasing the company employees satisfaction and motivation.

5.2 Future Work

Callbar Simulator has very potential in terms of features and scale. Some future work that could be developed is described below:

- Graphic Timeline improvement – hover information, icon labels, visualization of other events besides accepted and rejected calls, etc.
- Generate a report at the end of the simulation with the visualized analysis and possibility to download and later be used by data science teams.
- Incorporation of other metrics besides simple object quantities and times.
- Test the Callbar Simulator.

- Integration with presence API (from Talkdesk) and/or other modules – when possible test and scale the simulator to the whole product – full version of Callbar.
- Simulate different versions of the product and compare them.

Bibliography

- [1] [Online]. Available: <https://www.bestvalueschools.com/faq/what-is-a-capstone-project/>
- [2] [Online]. Available: <https://www.edglossary.org/capstone-project/>
- [3] [Online]. Available: <https://www.gartner.com/en>
- [4] [Online]. Available: <http://web.tecnico.ulisboa.pt/ist175564/landerz/>
- [5] [Online]. Available: <https://www.bestvalueschools.com/faq/what-is-a-capstone-project/>
- [6] [Online]. Available: <https://www.edglossary.org/capstone-project/>
- [7] [Online]. Available: <https://docs.talkdesk.com/docs>
- [8] [Online]. Available: <https://www.talkdesk.com/>
- [9] [Online]. Available: <https://academy.talkdesk.com/>
- [10] [Online]. Available: <https://socket.io/>
- [11] [Online]. Available: <https://www.w3schools.com/js/>
- [12] [Online]. Available: <https://www.interaction-design.org/literature/topics/user-research>
- [13] Mouhib Alnoukari, Moutasem Shafaamry and Kinaz Aytouni, "Simulation for Computer Sciences Education", Communications of the ACS, Vol. 6, N0.1 (2013)

[14] Rebecca Gill, Joshua Barbour and Marleah Dean, "Shadowing in/as work: ten recommendations for shadowing fieldwork practice", Department of Communication, Texas A&M University, College Station, TX, USA



Shadowing Log

Name: Agent 2 - (Talkdesk sales team)

Date: 12h15 to 12h45

General observations **[what he directly told us]:**

- Mean of 20 calls per day;
- Usually the calls are for company general numbers;
- Mean of 3min per call;
- User targets mostly Support, customer experience, VP operations and executive people inside the target possible company client;

General observations **[what we gathered from him]:**

- The user performed outbound call almost all times (given his junior role in the team);
 - Most of his time was spent on searching for potential clients;
 - User is always “Available” on Callbar;
 - User is 100% occupied with the call;
 - User does not perform multitasking (while on call);
 - User characteristics and Profile:
 - Persistent;
 - Calm;
 - Friendly;
 - Chill guy;
 - Junior worker (started 2 weeks ago);
 - Experiencing different methods to get his goal (possible clients);
 - Patient;
 - Does not care about rejection;
 - Retrieve fun from bad contact experiences and move on to the next contact;
-

(My Annotations):

Agent presence is “Available”;

Searched and got a potential client’s phone number;

Copied phone number and pasted it to callbar input bar;

Clicked on green phone button to initiate the call;

Call ringing... ; (outbound)

Waiting... 20sec ... ;

Nothing! No one answered;

Agent updated Sales Force (integration application) with notes (what happened, name, contacts, more info) written;

The agent did not give up after the previous unfortunate attempt to call someone;

Got the previous company’s general phone number and copied/pasted to callbar input bar;

Clicked on green phone button to initiate the call;

Call ringing...; (outbound)

Waiting... 25sec ...;

Someone answered;

The agent introduced himself in 5sec;

The agent asked if it was possible to speak to the previous person or someone equally responsible;

The agent said that he already talked to the first person in question the previous week (lied);

It was not possible to speak to anyone responsible but the agent got the direct email contact;

End greetings; [time of call - 2min +-]

The agent seemed pleased;

The agent updated callbar boxes after call ended;

Agent spent some time elaborating a good message to send; (3min +-)

The agent sent an email to the previous gathered email contact (to sell talkdesk's product and introduce it);

The agent put his regular workflow on standby to rest a little bit and at the same time started a conversation with me about his workflow and callbar usage; (5min +-)

The agent also asked specific questions about his work and features to a more experienced colleague; (2min +-)

The agent continued searching for new possible clients (in companies' websites, tech companies' lists, etc);

Agent wrote follow-up emails of contact;

He found a possible client (company);

Agent searched for specific positions and people responsible inside that company; (searched in app.datafox.com, etc);

Continued seeking for best people to contact;

The agent got the best contact he found;

Copied/pasted to callbar input bar;

Clicked on green call button;

Ringling...;

Person answered;

The agent performed an initial "sweet talk" to make the possible client comfortable and already liking him before he introduced the company and the product to sell;

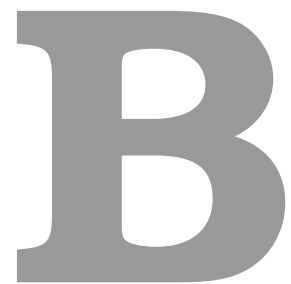
Agent asked for a call transfer to an inside target responsible and lead guy that he said he talked the

previous week (lied);
Apparently it was not possible to transfer the call to the desired person;
Then the agent asked for the direct contact;
Apparently it was rejected from the other side;
Then the agent asked (by curiosity) if they had another similar contact they could spare;
Request was rejected;
End greetings;
End call on red callbar phone button;
Info updated to callbar text boxes with gathered information (or lack of it);
The Agent did not seem pleased nor angry. It was a normal workflow;

The agent stopped his work for a little bit to rest and started talking to me again about his work, workflow, flaws, things he did and what methodologies he used and changed from last weeks (he started 2 weeks ago) until now;
The agent became less productive after his last call;

The agent got back to work and searched again for new contacts of possible clients;
He was constantly swapping between browser contact integrations and his work email;
Then the agent search for more contacts on previous browser lists and on LinkedIn;
Also searched for descriptions;

Searched for companies to contact. Found it;
Search for specific contact inside that target company; Found it;
Copied/pasted to callbar input bar;
Clicked on green call button;
Ringing... ; 14 sec... ;
Person answered;
The agent introduced the company and the product to sell very quickly;
The agent said he talked to another person last week (lied) and asked for the email contact because he only had the phone number;
He got the direct email;
Agent seemed pleased;
Then the agent pressed the red phone callbar button to hang up the call; [call duration - 25 sec];
Updated info on callbar text boxes;

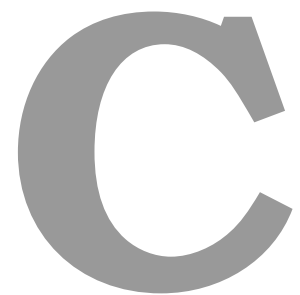


Interview Questions

Questions to ask:

1. Do you perform multitasking on Callbar?
 - If yes, how often?
 - Are you more or less productive? How so?
2. Do you perform Inbound and Outbound calls?
 - Which one do you do more? Ratio.
3. How many calls do you do per day?
 - Outbound vs Inbound
4. What is the mean duration of the calls?
 - Outbound vs Inbound
5. (Behaviour) How do you consider yourself?

- Patient/calm/friendly/objective...?
6. What external and internal problems of Callbar have you witnessed? (Callbar issues, internet connection lost, etc...)
 7. Tell us about call/message queueing.
 - How do you choose which team colleague gets to answer the call on wait?
 - Why?
 8. Open time to talk freely!



Talkdesk API (endpoints example)

This is an example for a list of users. The endpoint is '/users' and the HTTP method is 'GET'.
(Image C.1 in the next page).

HTTP	cURL
● 200 OK	<pre>GET https://api.talkdeskapp.com/users HTTP/1.1 Authorization: Bearer PGNsaWVudF9pZD46PGNsaWVudF9zZWNYZXQ+ Content-Type: application/json</pre>
	<pre>{ "total": 2, "page": 1, "per_page": 1, "embedded": { "users": [{ "id": "c65186ee-3780-468e-968b-ca15b981c312", "email": "john.smith@example.com", "name": "John Smith", "active": true, "gender": "Male", "extension": 123, "external_phone_number": "+18000000000", "created_at": "2016-06-14T16:23:07Z", "_links": { "self": { "href": "https://api.talkdeskapp.com/users/{user_id}" }, "account": { "href": "https://api.talkdeskapp.com/account" } } }], "links": { "self": { "href": "https://api.talkdeskapp.com/users" }, "account": { "href": "https://api.talkdeskapp.com/account" }, "next": { "href": "https://api.talkdeskapp.com/users?page=2" } } } }</pre>

Figure C.1: Example of Talkdesk API - input and output.



JSON File (example)

This is an example of a file that commonly interchanged between the front-end and back-end of the application.

This example is the 'Scenarios' file loaded in the setup phase of the simulation with the predefined scenarios.

```
1 var scenarios = [  
2   {  
3     "name": "Scenario 1",  
4     "number_calls": "10",  
5     "number_devices": "10",  
6     "call_duration": "300",  
7     "call_standby": "5000",  
8     "call_pickup_delay": "500",  
9     "profiles": [  
10
```

```

10         "shy",
11         "champion",
12         "normal",
13         "sleepy"
14     ]
15 },
16 {
17     "name": "Scenario 2",
18     "number_calls": "20",
19     "number_devices": "50",
20     "call_duration": "300",
21     "call_standby": "5000",
22     "call_pickup_delay": "250",
23     "profiles": [
24         "champion",
25         "champion",
26         "normal",
27         "normal"
28     ]
29 },
30 {
31     "name": "Scenario 3",
32     "number_calls": "35",
33     "number_devices": "12",
34     "call_duration": "500",
35     "call_standby": "5000",
36     "call_pickup_delay": "700",
37     "profiles": [
38         "shy",
39         "shy",
40         "sleepy",
41         "normal"
42     ]
43 },
44 {
45     "name": "Scenario 20",
46     "number_calls": "6",
47     "number_devices": "2",

```

```

48     "call_duration": "12000",
49     "call_standby": "8000",
50     "call_pickup_delay": "5000",
51     "profiles": [
52         "shy",
53         "shy",
54         "shy",
55         "shy"
56     ]
57 },
58 {
59     "name": "Scenario Simple",
60     "number_calls": "5",
61     "number_devices": "4",
62     "call_duration": "10000",
63     "call_standby": "5000",
64     "call_pickup_delay": "2000",
65     "profiles": [
66         "shy",
67         "normal",
68         "champion",
69         "sleepy"
70     ]
71 }
72 ]

```

