UNIVERSITY OF REGINA

ENSE 477: CAPSTONE PROJECT

REQUIREMENTS AND SPECIFICATIONS

Telport: Sasktel Telecommunications Portal

Authors
Dakota FISHER
Quinn BAST

Supervisor
Dr. Yasser MORGAN

February 4, 2019

Revision History

Revision Version	Revision Author	Revision Date
1.0	Dakota Fisher	February 4, 2019

Contents

List of Figures

List of Tables

1 Introduction

This document outlines and defines the requirements and specifications. The initial problem statement and proposed activities are suggestions given by Sasktel to provide a manageable scope. The decisions on infrastructure, aside from the Broadworks API and telephone account access are left up to us to decide.

1.1 Problem Statement

SaskTel requires a communications portal that can interwork with a Telephony Application Server and our core network to present communications and feature capabilities through a browser. This will allow for the exploration of new communications service models.

SaskTel has been pursuing the deployment of a new communications core along with the Cisco/Broadsoft Telephony Application Server branded as Broadworks. One of the drivers is to enable a richer customer experience through a converged architecture that exposes rapid development to enables new capabilities. Broadworks exposes the Application Programming Interface to access service control tools and user information. These tools and information can be used to create new communications applications or add additional value to existing applications.

The main objectives for SaskTel is to gain exposure to new and innovative communications service experience for our customers and to promote the potential internally for aligning resources, time and effort in enabling applications.

1.1.1 Proposed Activities

- Establish communications portal and platform interworking
- Gain a high-level understanding of the communications network architecture
- Gain an understanding of a method to access and use TAS APIs
- Establish base interworking and web browser communications portal
  - Access and exposure to TAS API
  - User registration through IMS
  - User presentation and interaction via portal
  - Exposure to 4-5 basic features to validate interworking i.e.
    (listed only as suggestions)
    - Call forwarding
    - Display of call logs (All, Incoming, Outgoing, Missed).
    - Simple call blocking by using a slider. Simple drop down to show numbers blocked (allow for unblocking).
    - Directory. Searchable by typing any string of characters.
- Enable WebRTC communications
- Enable the use of WebRTC for internet communications directly through the portal
- Create innovative communications experience
- Explore feature capabilities and experiment with innovative communications capabilities
- Demonstrate and showcase the ability to grow and share knowledge.

1.1.2 Skills Required

General Knowledge Requirements:

- Network Platform and service specific knowledge
- Software engineering skills
  - Client / server operation
  - IP Network Protocols (SIP, RTP, UDP, HTTP ...).
  - Internet Methods (Using XML, JSON, REST, ...).
  - Web Browser Programming (HTML, CSS).
  - Programming Languages (Java, Python, Ruby)

API Technology Comments:

- Most popular approach to delivering web APIs is REST (Representational State Transfer).
- API returns data in either XML or JSON.
- Most popular implementation is REST+JSON (not a standard but widely accepted within the industry).

API Usage Comments:

- Must consider how a resource will be manipulated not just retrieved.
- Should have strong understanding of both client-side and server-side programming.
- Should have strong understanding and experience with HTML and CSS (Cascading Style Sheets) for web programming, development, and design.
- Should have strong understanding and experience with Java Script.
- Should have a strong knowledge and experience with dynamic programming languages making Python and Ruby emerging industry favorites (Python+Django or Ruby+Rails).
- Should have knowledge and experience of the user and use of the product interface (in regard to forming the interface).

2 Design Decisions

During the creation of the project, we had full reign of the architecture and languages in which we wanted to use in order to create the solution for SaskTel. As a result, we did a significant amount of research on the different technologies available in order to make the best decisions for the project. While making these decisions, there were two different sides of the project to focus on. The backend would manage interfacing between SaskTel's servers as well as manage user credentials for the system, while the frontend would display the interface to the user. Both the backend and the frontend had may different decisions which led us to the current solution.

2.1 "Backend"

Some text

Date	Design Decision	Reasoning
Date	Design Decision	Reasoning
Date	Design Decision	Reasoning
Date	Design Decision	Reasoning
Date	Design Decision	Reasoning

2.2 "Frontend"

Some text

Date	Design Decision	Reasoning
Date	Design Decision	Reasoning
Date	Design Decision	Reasoning
Date	Design Decision	Reasoning
Date	Design Decision	Reasoning

3 Considered Frameworks

3.1 Node.js (In Use)

3.2 JavaScript

3.2.1 React (In Use)

3.2.2 Angular2 + Typescript (Not In Use)

3.2.3 Vue.js (Not In Use)

3.3 Python

3.3.1 Django (Not In Use)

3.3.2 Flask (In Use)

3.4 Test Frameworks

3.4.1 Mocha (In Use)

3.4.2 Jest (Not In Use)

3.4.3 Chai (In Use)

3.4.4 Sinon (In Use)

3.4.5 Enzyme (In Use)

3.4.6 Selenium (In Use)

3.4.7 Cypress (Not In Use)

4 Tools

4.1 Git

4.2 Toggl

4.3 GitKraken

4.3.1 Glo Boards

4.4 Google Suite

4.5 Discord

4.5.1 Alternative: Slack

4.5.2 Alternative: Skype

4.5.3 Alternative: Messenger

4.5.4 Alternative: WhatsApp

4.5.5 Alternative: Hangouts

5 Security Considerations

5.1 JSON Web Tokens

5.2 API Communication

5.3 Data Integrity

5.3.1

6 User Stories

6.1 Log In

6.2 Log Out

6.3 Check Call Logs

6.4 Turn on Call Forwarding Always

6.5 Turn on Call Forwarding Busy

6.6 Turn on Call Forwarding Selective

6.7 Turn on Call Forwarding No Answer

6.8 Turn on Call Forwarding

6.9 Turn on Do Not Disturb

6.10 View my Profile

6.11 Call a Phone Number

6.12 Start a Call to a Phone Number from my Phone

6.13 View Feature Access Codes