**Functional Requirements Document**

**Care Compare SK**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Description of Change** | **Author** | **Date** |
| 1.0 | Start of functional requirements doc | Kegan & Daris | Oct. 31st, 2019 |
| 1.1 | Added material to Methodology heading | Kegan & Daris | Jan. 7th, 2020 |
| 1.2 | Added diagrams and requirements information | Kegan & Daris | Mar. 3rd, 2020 |

CONTENTS

1 INTRODUCTION 4

1.1 Purpose 4

1.2 Scope 4

1.3 Background 4

1.4 References 4

1.5 Assumptions and Constraints 4

1.6 Document Overview 5

2 METHODOLOGY 5

3 FUNCTIONAL REQUIREMENTS 5

4.1 Context 5

4.2 User Requirements 5

4.3 Data Flow Diagrams 6

4.4 Logical Data Model/Data Dictionary 6

4.5 Functional Requirements 6

5 OTHER REQUIREMENTS 6

5.1 Interface Requirements 6

5.2 Data Conversion Requirements 7

5.3 Hardware/Software Requirements 7

5.4 Operational Requirements 7

APPENDIX A - GLOSSARY 11

1. **INTRODUCTION**

Project Vision: We want to make finding the best insurance smooth and easy for all types of people. We want to create an application that people of all different backgrounds and experiences actually want to use because it makes their life easier.

## Purpose

The purpose of this FRD is to organize and update the status of our capstone project over the next year. We created this project because we saw a need within society to provide a fast and efficient solution for finding great health care and this FRD helps bring the project to life and track its progress.

## Scope

This document will record every part of our capstone starting with requirements gathering, prototyping, diagrams and modeling, code, algorithms, testing, and anything else needed as the project proceeds. We are keeping the scope of the Capstone as small as we can simply because it is such a big topic and can become out of control easily. We are trying to only include a few insurance providers from Saskatchewan and a few key personas instead of focusing on everyone as a whole. This document will be updated regularly to make sure goals are being completed and that the project and document serve the purpose they set out to serve.

## Background

Our Capstone group consists of two people Kegan Lavoy & Daris Lychuk. We are working collaboratively on all portions of the project and are therefore responsible for the success of the project as one unit. We will both be adding to this document to keep us on track and organized, provide documentation for when tasks were completed, and give us a visual look at our Capstone at a high level.

## References

<https://www.thepolyglotdeveloper.com/2018/04/simple-user-login-vuejs-web-application/>

<https://www.policygenius.com/?fbclid=IwAR3u-OWY6MjbBXZi3Kf3_UNCH_z62TICDoBSemJ3GrBFbekl69dJpJxy1CA>

<https://www.canva.com/design/DADxkqME88I/dYV3DsKiU9dKoWcOuoFgpQ/edit?category=tACZChe47fQ#>

<https://www.webdesigndev.com/clean-modern-forum-templates/>

<https://www.webdesigndev.com/wp-content/uploads/2014/03/001507-Forum-Plus-Responsive-Drupal-Forum-Theme-RTL-Preview-ThemeForest.jpg>

## Assumptions and Constraints

[Provide a list of contractual or task level assumptions and/or constraints that are preconditions to preparation of the FRD. Assumptions are future situations beyond the control of the project, whose outcomes influence the success of a project.]

### Assumptions

-We are assuming we can get in touch with at least 2 insurance providers so that we will have real data to apply to our project.

-We are assuming that our users will try to access our site on a desktop/laptop instead of mobile device, therefore, will be designing Care Compare SK for optimal use on these devices.

### Constraints

- Time to produce the product

-Tech stack; need front end and back end that work together, database connection.

-No current API’s for Canadian insurance companies

-Insurance companies are private entities, not forthcoming with giving up their data easily.

## Document Overview

TBD

[Provide a description of the document organization.]

1. **METHODOLOGY**

**Agile Architecture**

We used an Agile architecture as a baseline methodology for our project. We did our work in 1 or 2 week sprints depending on the tasks being completed at that time. We chose this architecture because it allows for constant change and adjustments throughout a project which we find important as change will always be present.

**User story mapping**

We created epics and scenarios of the different tasks and functionalities our users would like to have when looking for insurance. This allowed us to visualize and keep track of what we need to accomplish as well as gather requirements for our application.

**Personas**

We created different categories of people to try and narrow down tasks that certain groups are more likely to do than other groups. This is also a type of requirements gathering as we are trying to get into the heads of different types of people to see what they are looking for within our application.

**Kanban board**

We used a Kanban board to organize and visualize our tasks that need to be done, are currently being worked on, as well as our completed tasks. This scheduling style works well within the Agile architecture because it can be constantly modified within the flow of the project.

**Low fidelity & high fidelity prototyping**

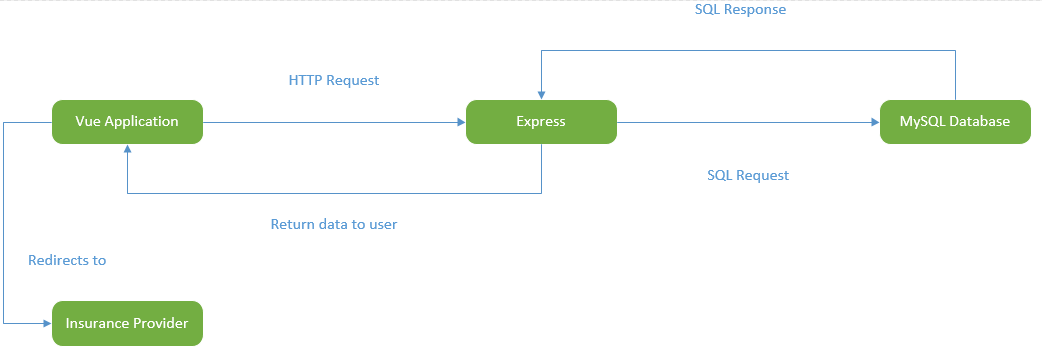
We used Adobe XD software to create low and high fidelity prototypes of our application. Adobe XD projects can be quickly and easily modified and changed which fits well with the Agile methodology.

1. **FUNCTIONAL REQUIREMENTS**

## Context/Communication

[Provide a context diagram of the system, with explanations as applicable. The context of a system refers to the connections and relationships between the system and its environment.]

Exhibit 2 - Generic Context/Communication Diagram

**

## User Requirements

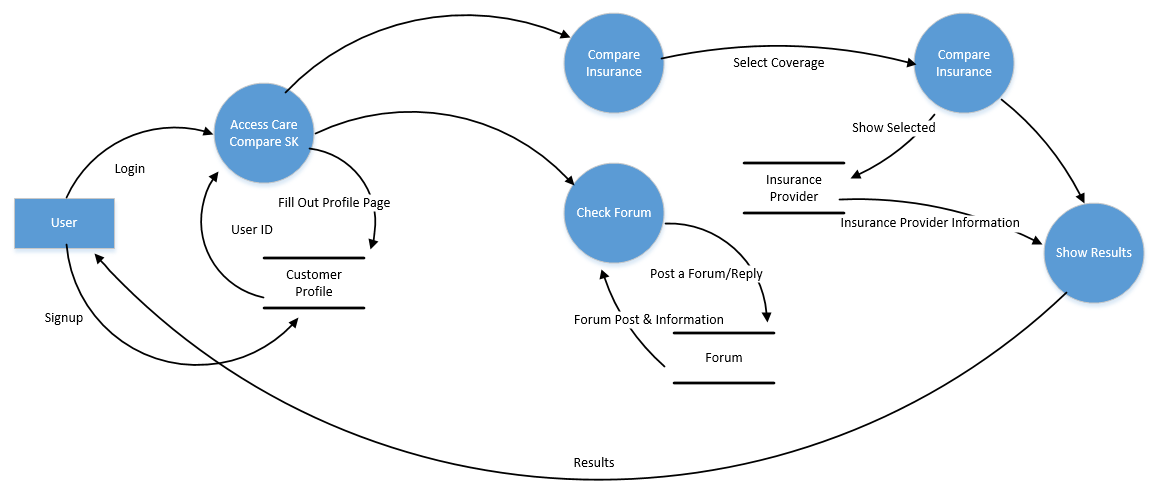
[Provide requirements of the system, user or business, taking into account all major classes/categories of users. Provide the type of security or other distinguishing characteristics of each set of users. List the functional requirements that compose each user requirement. As the functional requirements are decomposed, the highest level functional requirements are traced to the user requirements. Inclusion of lower level functional requirements is not mandatory in the traceability to user requirements if the parent requirements are already traced to them.

User requirement information can be in text or process flow format for each major user class that shows what inputs will initiate the system functions, system interactions, and what outputs are expected to be generated by the system. The scenarios should be comprehensive, to the extent that all user types and all major functions are covered. Give each user requirement a unique number. Typically, user requirements have a numbering system that is separate from the functional requirements. Requirements may be labeled with a leading “U” or other label indicating user requirements.]

1. All users are expected to be accessing *Care Compare SK* through either a desktop or laptop and not a smart phone, tablet, or other handheld device.
2. All users must provide a username and password in order to use *Care Compare SK*
3. All users have the option of providing additional information, such as, their full name, age, etc. This is not a requirement to use our website for the forum, however, the additional information is needed to use *Care Compare SK’s* comparison tool.
4. All users have the same security level whether full profile information is provided, or partial (username and password only).

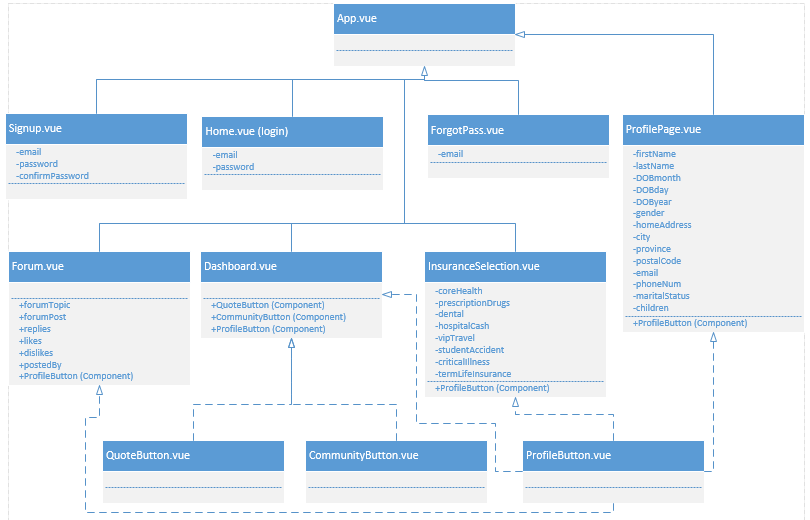
## Data Flow Diagrams

[Decompose the context level diagrams to determine the functional requirements. Data flow diagrams should be decomposed down to the functional primitive level. These diagrams are further decomposed during design.]



## Logical Data Model/Data Dictionary

[Create the initial Logical Data Model. Describe data requirements by providing data entities, decomposition, and definitions in a data dictionary. The data requirements describe the business data needed by the application system. Data requirements do not describe the physical database and are not at the level of identifying field names.]



## Functional Requirements

[List the functional requirements of the system.]

### Functional Requirements Group 1

[List the functional requirements for each functional requirements group.]

Exhibit 4 - Sample Requirements Group 1

|  |  |
| --- | --- |
| **Section/ Requirement ID** | **Requirement Definition** |
| FR1.0. | The system shall [parent requirement group 1]. |
| FR1.1 | The system shall [child/parent requirement]. |
| FR1.1.1 | The system shall [child requirement]. |
| FR1.1.2 | The system shall [child requirement]. |

### Functional Requirements Group 2, Etc.

1. **OTHER REQUIREMENTS**

[Describe the non-behavioral requirements.]

## Interface Requirements

[Describe the user interfaces that are to be implemented by the system.]

### Hardware Interfaces

[Define hardware interfaces supported by the system, including logical structure, physical addresses, and expected behavior.]

### Software Interfaces

[Name the applications with which the subject application must interface. State the following for each such application: name of application, external owner of application, interface details (only if determined by the other application).

It is acceptable to reference an interface control document for details of the interface interactions.]

### Communications Interfaces

[Describe communications interfaces to other systems or devices, such as local area networks.]

*Care Compare SK* is built on a local machine that is accessed via a port that was opened. The web application is completely local, other than when redirecting users to different insurance providers. This will direct users to the insurance provider’s own web application to find additional information about the specific insurance company.

## Data Conversion Requirements

[Describe the requirements needed for conversion of legacy data into the system.]

There should be no need to convert legacy data into the system.

## Hardware/Software Requirements

[Provide a description of the hardware and software platforms needed to support the system.]

All desktop/laptops should be able to use ANY web browser (*Google Chrome, Firefox, etc.)* to use *Care Compare SK*. Mobile devices, such as phones and tablets, may not work well with *Care Compare SK* styling. *Care Compare SK* should be able to run on any hardware, due to it not being a performance heavy web application.

## Operational Requirements

[Provide the operational requirements in this section.

Do not state how these requirements will be satisfied. For example, in the Reliability section, answer the question, “How reliable must the system be”? Do not state what steps will be taken to provide reliability.

Distinguish preferences from requirements. Requirements are based on business needs, preferences are not. If, for example, the user requires a special response but does not have a business-related reason for it, that requirement is a preference.

Other applicable requirements on system attributes may be added to the list of subsections below.]

Operational requirements describe how the system will run and communicate with operations personnel.

### Security and Privacy

[Provide a list of the security requirements using the following criteria:

A. State the consequences of the following breaches of security in the subject application:

1. Loss or corruption of data
   * Loss or corruption of data may lead to users needing to make a new account. It may also lead to losing valuable information on health insurance providers.
2. Disclosure of secrets or sensitive information
   * Disclosure of secrets or sensitive information would be the same as disclosure of privileged/private individual information. Our site does not necessarily contain any other secret or sensitive information.
3. Disclosure of privileged/privacy information about individuals
   * Disclosure of privileged/private information about individuals could lead to user information being sold or revealed publicly, such as their address, phone number, and date of birth. This would be extremely bad and potentially ruin *Care Compare SK*, as well as make our users lives more difficult.
4. Corruption of software or introduction of malware, such as viruses
   * Corruption of software or viruses could lead to false information to individuals. The viruses could be misleading and providing wrong information. If the viruses were to infect our users, it could potentially slow their systems down, or do other harmful things to their system.
5. State the type(s) of security required. Include the need for the following as appropriate:
6. Physical security.
   * No types of physical security are specifically needed currently.
7. Access by user role or types.
   * All users have the same security roles and privileges.
8. State access control requirements by data attribute. For example, one group of users has permission to view an attribute but not update it while another group of users has permissions to update or view it.
   * All users have access to edit their own information. As well as create a forum and/or reply to any forum already posted. Users do not have permission to delete other users’ posts.
9. State access requirements based on system function. For example, if there is a need to grant access to certain system functions to one group of users, but not to another. For example, "The system shall make Function X available to the System Administrator only".
10. State if there is a need for certification and accreditation of the security measures adopted for this application]

*The Security Section describes the need to control access to the data. This includes controlling who may view and alter application data.*

### Audit Trail

[List the activities recorded in the application’s audit trail. For each activity, list the data recorded.]

### Reliability

A. [State the following in this section:

1. State the damage can result from failure of this system—indicate the criticality of the software, such as:
2. Loss of human life
3. Complete or partial loss of the ability to perform a mission-critical function
4. Loss of revenue
5. Loss of employee productivity
6. What is the minimum acceptable level of reliability?

B. State required reliability:

1. Mean-Time-Between-Failure is the number of time units the system is operable before the first failure occurs.
2. Mean-Time-To-Failure is the number of time units before the system is operable divided by the number of failures during the time period.
3. Mean-Time-To-Repair is the number of time units required to perform system repair divided by the number of repairs during the time period.]

*Reliability is the probability that the system processes work correctly and completely without being aborted.*

### Recoverability

[Answer the following questions in this section:

A. In the event the application is unavailable to users (down) because of a system failure, how soon after the failure is detected must function be restored?

B. In the event the database is corrupted, to what level of currency must it be restored? For example “The database must be capable of being restored to its condition of no more than 1 hour before the corruption occurred”.

C. If the processing site (hardware, data, and onsite backup) is destroyed, how soon must the application be able to be restored?]

*Recoverability is the ability to restore function and data in the event of a failure.*

### System Availability

[State the period during which the application must be available to users. For example, “*The application must be available to users Monday through Friday between the hours of 6:30 a.m. and 5:30 p.m. EST.* If the application must be available to users in more than one time zone, state the earliest start time and the latest stop time. Consider daylight savings time, too.

Include use peak times. These are times when system unavailability is least acceptable.]

*System availability is the time when the application must be available for use. Required system availability is used in determining when maintenance may be performed.*

### General Performance

[Describe the requirements for the following:

A. Response time for queries and updates

B. Throughput

C. Expected rate of user activity (for example, number of transactions per hour, day, or month, or cyclical periods)

Specific performance requirements, related to a specific functional requirement, should be listed with that functional requirement.

### Capacity

[List the required capacities and expected volumes of data in business terms. Do not state capacities in terms of system memory requirements or disk space—if growth trends or projections are available, provide them]

### Data Retention

[Describe the length of time various forms of data must be retained and the requirements for its destruction.

For example, “The system shall retain application information for 3 years”. Different forms of data include: system documentation, audit records, database records, access records.]

### Error Handling

[Describe system error handling.]

### Validation Rules

[Describe System Validation Rules.]

### Conventions/Standards

[Describe system conventions and standards followed.

For example: Microsoft standards are followed for windows, Institute of Electrical and Electronics Engineers (IEEE) for data formats, etc.]

**APPENDIX A - GLOSSARY**

[Define terms, acronyms, and abbreviations used in the FRD.]