Swastha- The Health Monitoring System

**Software Requirement Specification**



***Objective:***

The aim of this software requirement specification document is to provide a thorough explanation of the specifications and features of our product to speed up user comprehension. This document will also aid the developers, designers, and stakeholders to have a clear knowledge about the final result of the product. At the same time, the developers will also be aware of possible challenges that can be faced during the development. Overall, for the project - Swastha, this document will serve as a development blueprint.

*Author*s :

|  |  |
| --- | --- |
| **Writers** | |
| Vishva Deliwala - 110091504  Kantat Rehnuma Noshin - 110091175  Jay Gabhawala - 110094322  Tahmid Abrar - 110090055  Swadhin Roy - 110094513  Rifat Rashedin - 110094216  Shivam Sunil Bhosale - 110090402  Khondoker Aminuzzaman - 110090822 | Written on: 15th November, 2022 |

*Sent to :*

|  |
| --- |
| **Destinataire** |
| Dr. Aznam Yacoub |

ID : Document3

References :

State : **Proposed**

# ***Introduction***

The developer's goal in this project would be to produce an inventive smart health monitoring system that considers the literary demands of the users. It should be designed so that users may check up on their health conditions on a regular basis. The smart health monitoring system allows an individual to closely monitor their changes in particular features like SpO2, heart rate, sleep cycle, BMI, step counts and maintain an optimal health status. Moreover, it will allow the user to be updated on the COVID 19 epidemic which will increase health awareness among themselves as well as around the world.

## ***Purpose***

The purpose of this document is to help all parties involved clients, product directors, developers, and testers to understand the fundamental concept behind the application, its prerequisites, and the key features needed to create the program.

While other users are advised to first understand the overall product before digging into specific functionality, developers and testers can concentrate only on the features.

## ***Document Conventions***

The SRS documentation followed accepted standard conventions. The text was bolded and reflected to draw attention to key elements. The finest SRS covers all hardware and software complexities simultaneously. A clear SRS paves the way for improved communication among developers and users and further empowers them to improve software development over time. Abbreviations are not of the utmost importance because the product's functional needs have been labelled with clear and intelligible names.

## ***Intended Audience and Reading suggestions***

The document outlining is to help all parties involved clients, product directors, developers, and testers to understand the fundamental concept behind the application, its prerequisites, and the key features needed to create the program.

While other users are advised to first understand the overall product before digging into specific functionality, developers and testers can concentrate only on the features.

## ***Product Scope***

Coding and testing for the product are presently underway. Coding has begun, and plans have been established to create weekly goals for monitoring and being updated also with operations and production heading forward. The scope of future expansion in this regard has been regularly monitored by the members.

The software is developed for many users. It will be accessible to anyone very easily. That way it will be easier to use this product and more users will be interested in using this. The main goal of this software is to improve quality of life. More about our goals is elaborated in our project proposal’s objectives section.

## ***References***

Information from:

1. [<https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>]
2. [<https://www.androidauthority.com/google-fit-393110/>]
3. [<https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for#jira-for-agile-teams>]
4. [<https://www.hitechwhizz.com/2020/08/6-advantages-and-disadvantages-drawbacks-benefits-of-https.html>]
5. [<https://www.atlassian.com/software/jira/guides/getting-started/overview>]

has been sourced for authoring this SRS in greater depth. Additionally, the SRS template's standard criteria have been properly followed .

# ***Overall Description***

## ***Product perspective***

Nowadays, we all rely heavily on our mobile devices today and enjoy learning about all kinds of things via them. Similarly, it is beneficial for us if we receive health-related information through our mobile device, and most people today want that. And because of this, they desire a mobile health monitoring system that would allow them to be informed of serious health issues earlier. So, they will not have to spend a lot of money fixing them.

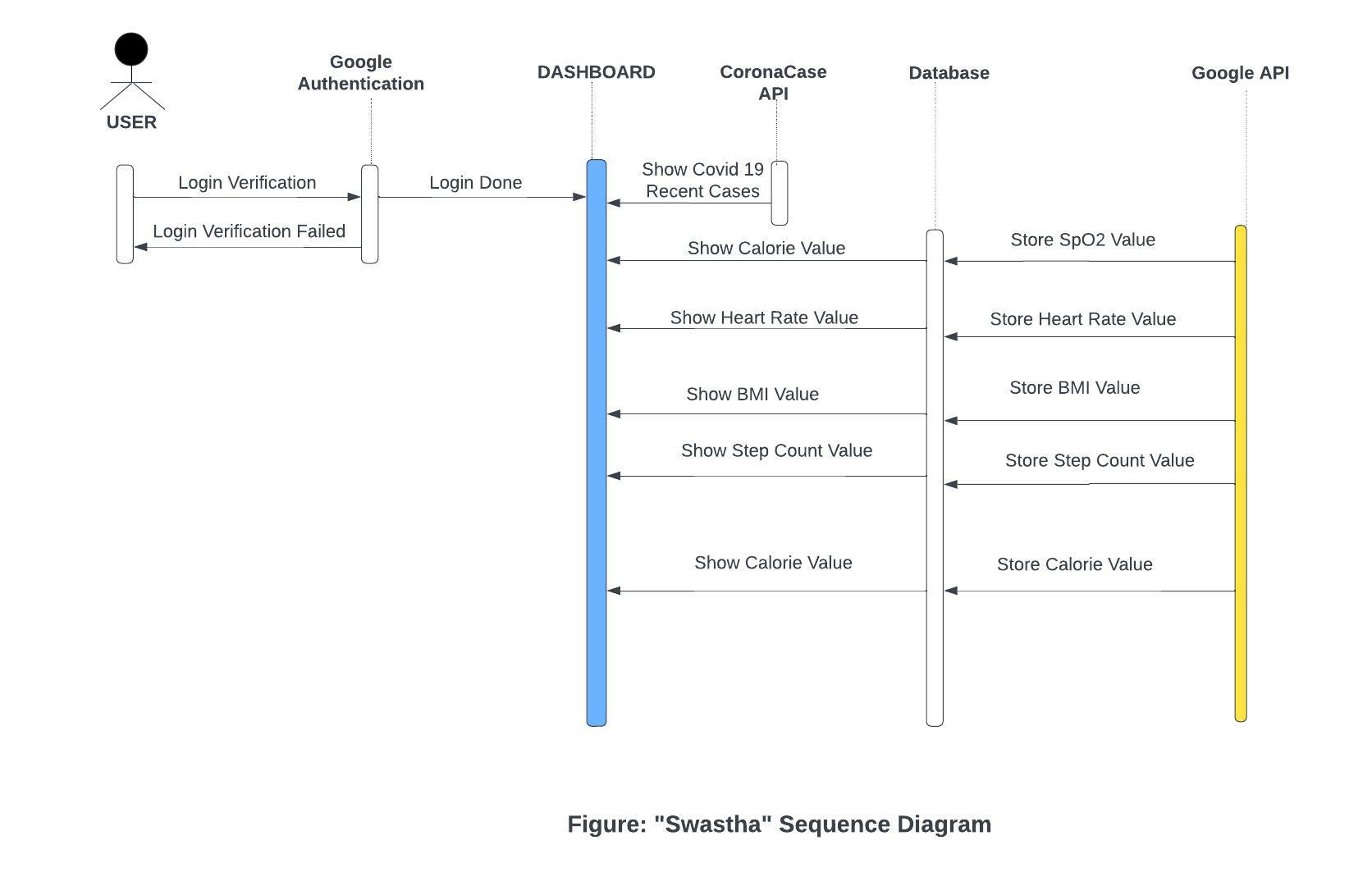
Our software will also be a platform where a user can easily know his body's essentials like calories, weight, BMI, heart rate, sleep cycle, oxygen level in advance. And from there he will get an idea whether he has a problem at all or needs a solution somewhere. In addition, our product will send an alert directly to the mobile if the rate of COVID 19 increases, through which everyone can easily know about the bad condition of COVID 19.

## ***Product functions***

The following are the main characteristics of the product:

* Sign up and login information.
* SpO2
* Heart beat
* Step counting
* Monitoring Sleep Cycle
* Calculating Body Mass Index
* Monitoring total COVID 19 cases per day
* Alert if there are too many COVID 19 cases. (1000-2500 thresholds)

## ***User classes and characteristics***



## ***Operating Environment***

Both Android and iOS devices will be able to run the app. It will support Android 11, also known as Red Velvet Cake internally, for Android devices. Most Android-powered devices run Android 11, which is the most recent version of the operating system. The software will support iOS v16 for iOS devices. Additionally, our application will work with Android and iOS later versions.

## ***Design and Implementation Constraints***

The design and deployment of the application will not be hampered by corporate or regulatory rules. As the product develops and has greater potential, security issues and programming standards will be examined in more detail. The coding languages Flutter and Dart, together with their libraries, represent the unique technology utilized. The Firestore database will be used, and other databases may be sourced as and when necessary.

## ***User Documentation***

After using the software properly, the first time, references to health concerns will be provided with the previously accessible data. While the product is being tested by the developers, there are also plans to make a video and user manual. The video will be produced in a way which is appropriately informative.

# ***External Interface Requirements***

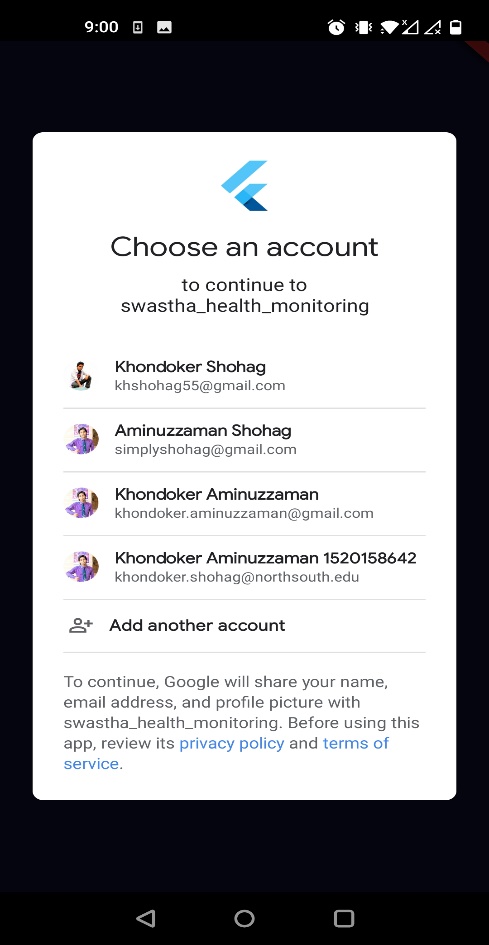
## ***User Interfaces***

Following will be the major user interfaces:

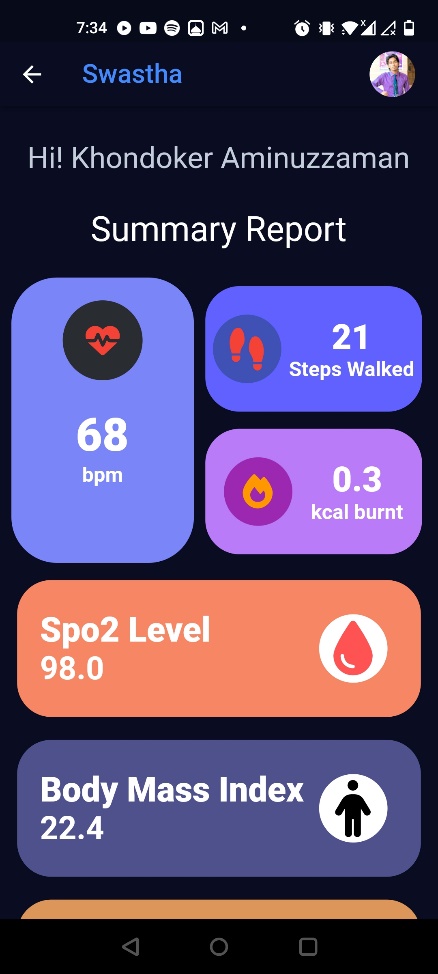
* ***Login Page***

******

* ***Google Login Page***

******

* ***Basic Dashboard***

******

* ***Profile Page***



* 



## ***Hardware interfaces***

The system requires a WearOS smartwatch with the integration of Google Fit and a smartphone to control the app. The app will fetch health related data from the user’s Google fit account, process it and show it on the user dashboard of the application.

* **Supported devices:** This is a Cross-platform solution which is compatible with Android and iOS at the same time. The application will be available on Play Store and App Store for the users to download and use for free.
* **Nature of data:** About the data, the app will collect quantitative real time data from the user’s Google fit account. This includes real-time numerical data of Heartbeat rate, SPO2 level, Step count and Sleeping hours.

## ***Software interfaces***

**Technologies and Tools:**

**Flutter:**

Flutter is an open-source software Development kit created by Google based on Dart programming language which is easy to learn and has cross-platform mobile development abilities.

**Why Flutter?**

* We want to target a larger audience and therefore with this cross-platform feature we would be able to make the app for both iOS and android devices.
* While there is React Native which is an alternative for Flutter, it seems that over time the React Native is projected to fall since historically Flutter has proven to give better performance than the latter.
* Flutter also has custom widgets which makes it easier to create great visuals for the application, while not worrying about UI compatibility on other devices.
* Flutter also comes with built-in theming support
* While developing, the hot rod features allows you to make the changes in the application in real time.
* A wide variety of apps like Airbnb, Uber, eBay are already running using flutter.

**Google Fit**

Google Fit is a health monitoring application created by Google which is compatible with Android, iOS, and Wear OS. For getting the health data from the user, we will be using google fit.

**Why Google Fit?**

* Google Fit offers free and open API to the developers, whereas the others do not.
* It is compatible with the Wear OS devices, and has all the health status parameters like heartbeat, SPO2 which we are looking to add to our dashboard.
* A non-essential feature could be that the google fit is available on a vast array of wearable devices thereby giving customers the benefit to choose one as per their budget and other requirements.

**JIRA**

Jira is a suite of agile work management solutions whose main task is to power collaborations across all teams. Jira helps teams plan, assign, track, report, and manage work and brings teams together for everything from agile software development and customer support to start-ups and enterprises. [1]

**Why JIRA?**

* Jira provides two types of hosting solutions: Cloud and On-prem. For this project cloud is a great fit as it lets us use it without any basic costs and device management risks.
* Jira uses some key terms which represent various entities and helps the team stay on the same page by defining them as follows:
* **Issue:** A Jira Issue refers to a task or work item that can be tracked from creation to completion.
* **Project:** Project is a collection of issues that have common context or purpose. This helps organize the issues by groups in the same project and set common rules.
* **Boards:** A board in Jira is a way to display issues in a project that makes it easy to manage and track. The Board is a visual representation of a team's workflow within a project [1].
* Jira uses some workflows which represent the state of the Issue and the states it can move to from the current state. Examples of states can be NEW, IN PROGRESS, IN REVIEW, CLOSED.
* All things considered, Jira is one of the best management tools that lets us organize, divide, manage, assign, track and collaborate the tasks of a project within a team which is why it is used by the majority of the Software and IT Industry.

**GitHub:**

GitHub is a version control system.  It is a web-based Git repository hosting service that offers the functionality and features of Git's distributed revision control and source code management (SCM).  The main reason to use Git is to keep track of every change in the project as you do. GitHub is basically a hosting service.

**Why GitHub?**

* Amazing system for the open-source codebase.
* Allows easy sharing.
* Offers pull requests and comments.
* Easy setup.
* Allows remote collaboration.
* Provides easy control features.
* Has a consistent addition of new features.

**Firestore(Firebase):**

The Firestore database is the most recent one used by Firebase to create mobile app. Firestore will be used as the primary database to store user data for this software. For this system, the user's heartbeat rate, SPO2 level, step count and sleeping hours of the last 7 days will be stored in the database. This data will be used further to create a weekly update on the user's health condition. The data collected for the Covid-19 cases will also be stored here and the data will be used to create comparative graphs regarding the situation of Covid-19 cases. Moreover, a unique ID will be created for each user to store the data will be stored separately in the database.

## ***Communication interfaces***

An active Internet connection, whether through WIFI or Cellular data will be required to access and fetch data from Google Fit API and Covid-19 Canada API. HTTP protocol will be used for communication over the internet. The data encryption provided by HTTPS is one of its main advantages. Asymmetric and symmetric encryption are both used to safeguard the confidentiality and integrity of data while it is in transit. A client and a server establish a secure session using asymmetric encryption, and during the secure session, data is exchanged using symmetric encryption. Over an unsecured network, HTTPS establishes a secure channel. If sufficient encryption suites are used and the server certificate is validated and trusted, this guarantees a decent level of protection against eavesdroppers and man-in-the-middle attacks. We also connect to a REST API using HTTPS protocol to secure the connection by providing the same encryption techniques. In this way, users can ensure that their information is transmitted to the proper location and not to any phishing frauds.

# ***System Features***

## ***System Feature 1***

**Home Page Dashboard Panel:**

This is the main door way for the user to access all other features of this system. Therefore, it has the highest priority among the other features. The home dashboard provides real-time access to all available health related information of the user and the covid-19 related updates. The dashboard has all the panels which will directed to each feature, for instance: Heartbeat panel, SPO2 panel, Step counter, Sleep Cycle, BMI and Covid-19 Update.

## ***System Feature 2***

**Heartbeat panel:**

This screen displays the heart rate in beats per minute(bpm). The heartbeat page will display:

* Real time heartbeat
* Heartbeat history of the last seven days
* Any inconsistencies in heartbeat
* Information about various heartbeat rates
* Advice on how to keep the heartbeat rate normal.

There is also a setting option where the user can add a range of heartbeats where an emergency alarm can be triggered, and the app can send an emergency message to the emergency contacts they have designated.

## ***System Feature 3***

**SPO2 panel:**

This panel displays the SPO2 levels in real time. In the SPO2 level page, a user will find:

* Real time SPO2 level
* A history graph of the previous seven days SPO2 levels
* The normal and abnormal range of SPO2 level which helps the user to understand any abnormality in the user’s SPO2 level
* Any important and relevant suggestions about how to maintain normal SPO2 level.

There is also a setting option where the user can specify the SPO2 level range at which an emergency alarm can be triggered and the app can send an emergency message to the emergency contacts they have designated.

## ***System Feature 4***

**Step counter:**

This panel shows the daily steps taken, the status of the objective, and if it has been attained. There are three main pieces of information that will be displayed on the step counter page:

* The number of steps taken and the remaining distance to the target.
* The distance that has been traveled in kilometers(km).
* Amount of calories burnt while walking.

With the Steps history option, all three criteria will be displayed for both the current day and the last seven days.

## ***System Feature 5***

**Sleep Cycle:**

This panel shows the sleep time of the user. The Sleep Cycle section will include data on:

* User’s sleeping time in hours(hr)
* Any goals or accomplishments related to sleep cycles for both the current day and the previous 7(seven) days

Users will be able to set targets for their sleeping cycles using the set goal option, much like they do with the step counter feature. They can also track their sleeping history.

## ***System Feature 6***

**BMI score:**

This panel shows your current BMI (Body Mass Index) in comparison to the goal of the user. In the BMI section, the app will provide:

* Current BMI
* Weight
* Height
* Any change in their BMI

The user will manually put their current weight from the weight and height in the system, and the system will provide the user's BMI score.

## ***System Feature 7***

**Covid-19 Update:**

This panel shows the update regarding covid-19 cases. It will provide,

* Real time covid-19 cases
* A graph containing the cases for the last seven days
* Notification to the user’s phone if the cases are more than 2500 per day

# ***Non-Functional Requirements***

## ***Performance Requirements***

Our main goal for this project is to give our customers a healthcare system that will help them examine their health condition continually. In our final product, the end users will have a system which is cost efficient, easily accessible, and very user friendly. There are a few performance requirements that we must maintain to make the users satisfied with the service it will provide. As non-functional requirement implies features through which issues in the health-care framework can be resolved, these performance requirements are needed for the product to run efficiently for the end users and they are:

* Connectivity
* System Reliability
* Higher Efficiency
* High Speed
* Ambient Intelligence
* Quality of Service
* Memory
* Interoperability

## ***Safety Requirements***

Some criteria are focused on potential loss, injury, or harm that may arise when using the product. The possible risks that can arise during the project development are given below:

* **Dependent Risk:** The project is dependent highly on Google Fit API for the data gathering task. In the unlikely event that this API is changed or the service is down it can affect the product and would require a maintenance release to make it compatible with the newer versions of the API.
* **Unpredictable risks:** Additionally, there can be some unpredictable risks which should be considered. Incidents like, alterationst5 s in the law, changes in the economy can be considered as unpredictable risks that can hamper the project development.

## ***Security Requirements***

Smart healthcare becomes more vulnerable to threats even though it helps provide better healthcare to people all over the world. The security requirements in smart healthcare systems vary from the traditional security techniques at times due to the dynamic nature and smaller form factors. As the healthcare networks contain personal information, we must fulfill the basic security requirements so the information cannot be easily tampered with which is mentioned below:

* Authentication
* Location Privacy
* Data Confidentiality
* Access Control
* Data Freshness
* Data Privacy
* Service Availability
* Unique Identification
* Resilience

## ***Software Quality Attributes***

**Usability:**

For any user to understand the user interface, it must be straightforward and unambiguous.

**Availability:**

The software must to be accessible always. To offer a better customer satisfaction, it should be made sure that there is little to no downtime.

Additionally, the software needs to be stable and strong. It ought to produce accurate outcomes. Additionally, the system must make sure that, when a user requests an update, the proper message reaches its intended recipient without being lost or altered.

**Testability:**

The software ought to be tested. The application must be tested in a different paradigm so that quality assurance engineers may look for faults or missing criteria.

**Maintainability:**

The software need to be created in a way that allows for future modification and expansion. A modification in the current requirements or the integration of new functionality ought to be simple.

## 

# ***Market Analysis***

After doing some research about the existing similar software, we have found some solutions such as Apple Health and Samsung Health which work on monitoring health parameters of the users.

How existing similar software's work till date:

* Most of the renowned software like Apple Health and Samsung Health are only compatible with their own environments.
* The existing apps only provide health-related user updates. There is no option to send any emergency alerts in case of any abnormal health conditions.
* None of them have any covid-19 related update system.

How Swastha shall work:

* Unlike the other existing solutions, the user would be able to connect our application with any brand's device which has WearOS or Google Fit compatibility.
* The next feature which makes it a unique one is the ‘Emergency Help’ feature. Based on the live vitals of the user, our system would be able to detect any anomaly and would be able to send an SOS to emergency contacts. There would be a short window where the user could refuse to do so in case of a false alarm and set the range for which the user wants to send an emergency alarm.
* Another useful feature is that the app will provide update regarding covid-19 cases across Canada. The data will be updated in real time. Moreover, the data of daily cases will be presented using a graphical representation which will allow the user to understand the situation more accurately.
* If the cases are more than 2500 per day, which is quite alarming, our application will notify the user by sending an alarm notification to their phone.

***Cost Analysis***

* Single Resource Cost: $45/hr. (considering average software engineer pay in Canada)
* Working hours per week: 30
* Cost for 5 Resources: $54,000/ 3 months
* Jira: $14 per user/month
* GitHub (Version Control System): $21 per user/month
* Maintenance: $500-$12,000/year (Regular updates and fixes)