Project Plan – Milestone 1

Mobile Team

Presented by: Sylvia Lopez, Bertina Lee, Brian Malott, Karim Mansour, Komi Noukafou, Sompon Boontho

SWEN 670 – sOFTWARE eNGINEERING pROJECT

February 2, 2021

reVision 1.2

Project name: Form Scriber, AI Mobile Application

Date: January 13, 2021

Project leader: Sylvia Lopez-Willis

Phase: Project Planning

For approval: Sylvia A. Lopez-Willis

Sylvia A. Lopez-Willis Date: 2/2/2021

For approval: Dr. Mir Mohammed Assadullah

Date: 2/2/2021

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Date | Description | Approved By |
| 1.0 | 1/13/2021 | Initial Project Plan Release | Sylvia Lopez-Willis |
| 1.1 | 2/23/2021 | Updates to the Project Plan and Software Test Plan added. | Sylvia Lopez-Willis |
| 1.2 | 3/16/2021 | Updates to the Project plan and Software Test Plan. | Sylvia Lopez-Willis |

Table of Contents

[General information about the project 5](#_Toc66722962)

[1. Situation sketch and problem definition of the project 5](#_Toc66722963)

[2. Project assignment 6](#_Toc66722964)

[2.1 Project Scope 7](#_Toc66722965)

[2.2 Project Assumptions 8](#_Toc66722966)

[2.3 Product Approval/Acceptance Criteria 8](#_Toc66722967)

[2.4 Project Exclusion 9](#_Toc66722968)

[2.5 Acronyms and Abbreviations 9](#_Toc66722969)

[2.6 Referenced Documents 9](#_Toc66722970)

[3. Risk Analysis 10](#_Toc66722971)

[3.1 Risk Matrix 11](#_Toc66722972)

[3.2 Analysis 11](#_Toc66722973)

[3.3 Risk Trigger 13](#_Toc66722974)

[4. Organization of the project 13](#_Toc66722975)

[4.1 Communication 14](#_Toc66722976)

[4.2 Explanation of Agile methodology using Scrum framework that will be applied 14](#_Toc66722977)

[4.3 Project Staffing 15](#_Toc66722978)

[4.4 Roles and Responsibilities 16](#_Toc66722979)

[4.5 Responsibility Assigned Matrix (RACI) 18](#_Toc66722980)

[4.6 Project Tools 21](#_Toc66722981)

[4.7 Project Document Storage 21](#_Toc66722982)

[4.8 Deliverables 21](#_Toc66722983)

[4.9 Initiation phase 21](#_Toc66722984)

[4.10 Execution Phase 22](#_Toc66722985)

[4.10.1 Change Management Plan 22](#_Toc66722986)

[5 Cost and Schedule Overview 23](#_Toc66722987)

[5.1 Cost Overview 23](#_Toc66722988)

[5.2 Schedule Overview 23](#_Toc66722989)

[6 Software Test Plan 23](#_Toc66722990)

[6.1 Test Plan Identifier 23](#_Toc66722991)

[6.2 Introduction 24](#_Toc66722992)

[6.2.1 Background 24](#_Toc66722993)

[6.2.2 Objective of the Test Plan 24](#_Toc66722994)

[6.2.3 Objective of STPMA 25](#_Toc66722995)

[6.3 Reference Documents 25](#_Toc66722996)

[6.4 Test Items 25](#_Toc66722997)

[6.5 Feature to be tested 31](#_Toc66722998)

[6.6 Features not to be tested 31](#_Toc66722999)

[6.7 Approach 31](#_Toc66723000)

[6.7.1 Develop Tests 32](#_Toc66723001)

[6.7.2 Prepare to Test 33](#_Toc66723002)

[6.7.3 Run Tests 33](#_Toc66723003)

[6.7.4 Review Test Results 33](#_Toc66723004)

[6.7.5 Test Results and Defects 34](#_Toc66723005)

[6.8 Item Pass/Fail Criteria 36](#_Toc66723006)

[6.9 Test Plan Matrix 37](#_Toc66723007)

[6.10 Test Deliverables 37](#_Toc66723008)

[6.11 Testing Tasks 37](#_Toc66723009)

[6.12 Environmental Needs 38](#_Toc66723010)

[6.13 Responsibilities 38](#_Toc66723011)

[6.14 Staffing and Training Needs 38](#_Toc66723012)

[6.15 Schedule 39](#_Toc66723013)

[6.16 Risks and Contingencies 39](#_Toc66723014)

[6.17 Test Revision History 40](#_Toc66723015)

[References 41](#_Toc66723016)

[Appendices: 42](#_Toc66723017)

[Appendix A - WBS 42](#_Toc66723018)

[Appendix B – Schedule of Deliverables 47](#_Toc66723019)

[Appendix C – Project Timeline 47](#_Toc66723020)

# General information about the project

# 1. Situation sketch and problem definition of the project

A. Background

1. The Form Scriber mobile application project concept was created by Dr. Mir Assadullah for the UMGC Software Engineering Project course Capstone Project. The Mobile Application team will be working with the Dialog flow Team, and DevSecOps Team to complete the overall Capstone Project. After the Capstone Project, the customers (public service professionals) will be able to download the application for use.
2. The Capstone Project's customers include public service professionals such as doctors, medical technicians, police officers, and others that would benefit from the service.
3. The Mobile Application Team will be working on the Form Scriber remotely. Teams have been assigned based on skill and background chosen by UMGC Faculty staff.

B. Statement of Need

1. Public service professionals need a mobile application that will allow them to prepare a text report.
2. Presently, no AI mobile application using chatbot has been created to prepare a text report for public service professionals required to document a portion of their job.

C. Vision Statement

1. This mobile application will be the first AI Application to use chatbot and offer public service professionals an application that will listen to a conversation between an interviewer and interviewee. Then, the mobile app will capture the conversation as a recording and send the captured recording to the Dialog Flow AI for processing the voice recording. A completed form with data collected will be offered by the Dialog Flow AI. The Mobile app will generate a professional summary report based on the form template. Finally, a report is generated for the customers to view and open with the option to print.
2. Contractors for the mobile application will be graduate students attending the Software Engineering Project course to complete the Masters of Science in Information Technology with Software Engineering Specialization, Capstone Project.

# 2. Project assignment

The project goals and objectives are described in detail in the Table 1 below. Using the SMART tool that stands for S-pecific, M-easurable, A-chievable, R-elevant/Realistic, and

Time-bound, the table has separated SMART to give the reader a more organized view of the goals and objectives using SMART (Project-Managment.com, 2019). SMART is written to the left columns, and the row to the right of the corresponding column answers the particular goal. Starting with Specific will answer what the project is trying to accomplish to answer the questions who, what, why, where, and which (Project-Managment.com, 2019). Measurable, will address how the project goal will be tracked for the development of the mobile application, and know how to measure it (Project-Managment.com, 2019). Achievable, will address the objective of how the project can be achieved, can the members of the team achieve the objective, and are we equipped with the skills and knowledge to reach our goal (Project-Managment.com, 2019). Realistic/Relevant is defining that the goals and objective are realistic, and that objectives coincide with the overall goal for the whole project (Project-Managment.com, 2019). Time-bound address the objective and goal deadline (Project-Managment.com, 2019). Overall, the SMART tool sets a guideline for addressing goals, and objectives in the project.

Table 1- SMART Goals and Objectives

|  |  |
| --- | --- |
| **Specific** | Currently, public service professionals do not have an AI mobile application that will allow them to prepare a text report on their Android or iOS mobile devices. The Form Scrober mobile application will listen to a conversation between an interviewer and interviewee (example: doctor and patient), capture the conversation as a recording, and send the captured recording to the DialogFlowAI for processing the voice recording. Once the DialogFlowAI processes the request, a generated professional summary report based on the selected preset/customized form template will be created. Upon completing the generated report, a link to the users Google Drive account will allow the user to view the completed PDF report, and print it if they would like through Google Drive. The mobile application will have three components, a recorder, a user interface, and a back end.  The mobile application will be available on the Google Play Store for Android and the App Store for iOS mobile devices. |
| **Measurable** | To develop the mobile application, an Agile methodology will be used with the Scrum framework. At the end of every Sprint, there will be a prototype shown at each Milestone meeting. |
| **Achievable** | According to skill and experience, graduate students in the MSIT course Software Engineering Project have been assigned to roles by faculty staff. The Mobile Team consists of four Developers, one Tester, and a Project Manager, requiring a strong team to achieve the goal of completing the project in all its phases. |
| **Realistic** | The project will move forward and has been assigned by the Software Engineering Project professor Dr. Mir Assadullah. The Form Scriber mobile application will be offered to public service professionals upon completing the Software Engineering Project course's eleventh week. |
| **Time** | The mobile application will be launched in eleven weeks from the first day of class, January 13, 2021. Using Agile with the Scrum framework, a prototype will be shown before each Milestone delivery date. Upon completing the mobile application testing phase, the system will be launched for Android and iOS mobile devices for free to all public service professionals through the Google Play Store and App Store. |

The Mobile Team is committed to delivering a quality application on budget and on time. We will not compromise on quality and add features that are not necessary. Waste will not be an issue.

## 2.1 Project Scope

The Project Plan's scope for the Mobile Team starts with an Initiation Phase, where the goals and objectives are identified, requirements are obtained, a Statement of Work (SOW) is created, the project's scope is determined, and communication is planned. The project schedule has been completed and can be viewed in a more detailed breakdown in [Appendix A](#AppendixA).

The project's work scope will be to create an artificial intelligence (AI) mobile application for customers geared toward doctors, medical technicians, police officers, and others in similar fields. The mobile application will be created in Android Studio using Flutter, a Google UI toolkit for creating an Android and iOS mobile application. The Dart programming language developed by Google to be used with Flutter will use one programming language to create the mobile application as a cross-platform for iOS and Android.

The mobile application will listen to the conversation between the customer and their patient (interviewee) and prepare notes that the customer will then be able to edit before saving it in an Electronic Medical Record or another similar tool the customer uses.

The mobile application will use a tool to guide the customer through how they want to see the end report. The device will create a template that can be used later during the customer conversation with the interviewee. The template will have multiple elements that the Artificial Intelligence (AI) part of the mobile application will work on and fill in the required information as the conversation progresses.

The Agile methodology will be used with the Scrum framework, because of the short time frame of the project and the number of developers on the team this framework was the best option. [Section 4.2](#_4.2_Explanation_) goes into more detail, and explanation of the methodology and framework.

Work within the scope of the project and what will be done:

* The mobile application (component) will do the sound capture and participate in processing.
* Google Cloud Platform (GCP) and its AI tools will be used. A new sign-up with GCP will be done that gives a $300 credit to use within three months.
* A chatbot service called Dialogflow offered through GCP will be used to power the system.
* Testing will be done on the mobile application to ensure it is ready before the launch date (completed by the set Capstone Project deliverable date).

## 2.2 Project Assumptions

The project assumptions have been made and described in the bullet points below.

* The scope will not be adjusted unless the Stakeholder, Dr. Mir Assadullah, changes the requirements or approves a change.
* The mobile application will be available for Android and iOS mobile users through the Google Play Store and App Store for free.
* Team members will not move from their roles unless approved and based on project needs.
* The health of the Project will be posted to the Professor's Office in LEO every Sunday for the life of the project to monitor the budget and health of the project.
* All deadlines for Milestone deliverables are set dates and will not be changed.
* Because of the project's short timeline, the execution of sprints will be held weekly for six weeks.
* All changes will be logged in a Change Log and approved by stakeholders.
* Because there is a potential, the mobile application will capture personal identification information, and each page will identify PPI and HIPPA.
* The software will be available for free to the public and have no warranty, and is provided “AS-IS.” The University of Maryland is not responsible for any shortcomings, and the user is solely responsible for the use.

## 2.3 Product Approval/Acceptance Criteria

The project is broken down into Milestones that will have set deliverable dates. Milestones will be broken up into four parts, and the Stakeholder, Dr. Mir Assadullah, will grade each Milestone. If Milestone 1 that includes the Project Plan and Software Requirements Specification, needs to be changed, the update will be included in the following Milestone deliverable for approval/acceptance.

This document contains a Change Management Plan that will go into more detail for any changes that need to be made. Changes will require approval by the Project Manager and Stakeholder, a Change Log will be kept and managed through MS Teams to track changes and approval/acceptance of those changes.

## 2.4 Project Exclusion

This section lists what is out of scope for the project.

* No AI models will be developed.

## 2.5 Acronyms and Abbreviations

|  |  |
| --- | --- |
| Acronyms and Abbreviations | Definitions |
| AI | Artificial Intelligence |
| GCP | Google Cloud Platform |
| POC | Point of contact for the developers on the mobile team between other channel team developers. |
| PII | Personal Identification Information |
| HIPAA | Health Insurance Portability and Accountability Act |
| SMART | SMART tool that stands for S-pecific, M-easurable, A-chievable, R-elevant/Realistic, and T-ime bound. Used to define project goals, and objectives. |
| STPMA | Master Software Test Plan for Mobile Application. |

## 2.6 Referenced Documents

Table 2 below shows documents that were used as a reference for this Project Plan. The Title column shows the title of the document, and the Reference column corresponds to the appropriate title where and who the document was referenced. For example, the reference lists website locations and the name of the author of the document.

Table 2 - Referenced Documents

|  |  |
| --- | --- |
| Title | Reference |
| The 2020 Scrum Guide | <https://www.scrumguides.org/scrum-guide.html#increment> |
| Kick-Off Meeting, SWEN 670, Software Engineering Project, Course Homepage | <https://learn.umgc.edu/d2l/home/545048> |
| Software Requirements Specification | Sylvia Lopez-Willis |
| Top five causes of scope creep… and what to do about them | Project Management Institute. https://www.pmi.org/learning/library/top-five-causes-scope-creep-6675 |
| Risk analysis and management a vital key to effective project management | https://www.pmi.org/learning/library/risk-analysis-project-management-7070 |
| Health Insurance Portability and Accountability Act of 1996 (HIPPA) | https://www.cdc.gov/phlp/publications/topic/hipaa.html |
| What is personally identifiable information (PII)? How to protect it under GDPR | https://www.csoonline.com/article/3215864/how-to-protect-personally-identifiable-information-pii-under-gdpr.html#:~:text=Personally%20identifiable%20information%20(PII)%20is,to%20identify%20a%20specific%20individual.&text=The%20GDPR%20also%20requires%20that,more%20specifics%20on%20the%20regulation.) |
| The Risk Management Process in Project Management | https://www.projectmanager.com/blog/risk-management-process-steps |
| Project Management Handbook | Project Management Handbook. Through the LEO Course Introduction Page.  https://learn.umgc.edu/d2l/le/content/545048/Home |
| How to Use the Risk Assessment Matrix in Project Management | DZone > Agile Zone  https://dzone.com/articles/how-to-use-the-risk-assessment-matrix-in-project-m-1 |

# 3. Risk Analysis

The below Risk Analysis in Section 3.2, [Table 1](#Table1) is broken up into risk, trigger event, responsible department, consequence, mitigation. A Risk Matrix in Section 3.1 is displayed as a reference. Six steps of the Risk Management Process have been implemented into this project that includes the following:

1. Identify the Risk
2. Analyze the Risk
3. Prioritize the Risk
4. Assign an Owner to the Risk
5. Respond to the Risk
6. Monitor the Risk

The risk will be identified during the Sprint Planning and weekly meetings. An analysis of the risk will be done, and how to resolve or minimize the risk (Ray, Stephanie, 2017). [Table 1](#Table1) shows a Risk Analysis of potential risks. Prioritizing the risks by severity and addressing the risks that require urgent attention will be discussed in the Sprint Meetings and as we move forward with the Sprint Planning. A Risk Matrix can be seen in Section 3.1. Assignment of the risk to someone on the team will be done during the weekly Sprints, based on what the team member is working on and its relation to the risk. The team as a collective will work together to respond and address the risk. Monitoring the risk will be done by the team member tasked with owning the risk, and the owner will discuss the risk during Sprint Meetings and planning (Ray, Stephanie, 2017).

## 3.1 Risk Matrix

The risks are listed in priority from low to high that includes a likelihood it could occur. This Risk Matrix will be used as a determination of how severe the risk is. The Risk Matrix will be referenced at the onset of the risk detection and communicated to the team, so the proper owner of the risk will prioritize how the risk will be addressed.

Table Risk matrix model to categorize project risks.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Likelihood |  |  |  |  |
| Consequence | 1 | 2 | 3 | 4 | 5 |
|  | Rare | Unlikely | Possible | Likely | Almost Certain |
| 5 Catastrophic | 5 | 10 | 15 | 20 | 25 |
| 4 Major | 4 | 8 | 12 | 16 | 20 |
| 3 Moderate | 3 | 6 | 9 | 12 | 15 |
| 2 Minor | 2 | 4 | 6 | 8 | 10 |
| 1 Negligible | 1 | 2 | 3 | 4 | 5 |

## 3.2 Analysis

Risk plans have been put in place so Development, Testing, and Project Manager will be prepared for the risk.

While these risks are planned, as the project progresses, more risks will be added to this analysis report throughout the project's life.

Table 4 - Risk Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Risk # | Risk | Trigger Event | Responsible | Consequence | Mitigation |
| RI.1 | Not able to deliver a working product | 5 | Developers, Testers | Mobile app crashes at startup, intermittent crashes, lags, does not meet customer requirements. | A prototype will be created before the launch that will be tested by the team and shown to a customer. Feedback will be used to make necessary changes or add features. |
| RI.2 | Audio recording limitations with Flutter. | 1 | Developers, Testers | The mobile application is not able to capture a conversation recording because of Flutter limitations. | Research, develop, and test Flutter Sound Plugin for audio recorder and player before launch. |
| RI.3 | Dependency on an unfamiliar language | 5 | Developers | Development takes longer. Prone to numerous defects. | Developer(s) work closely with other developers with experience in the language. Developers will utilize tutorials to become familiar before the start of development. |
| RI.4 | Scope Creep | 4 | Project Manager, Developers | Unauthorized changes or features.  Causing approved features to be left out of the product. Thus, creating an application lacking requirement features. | Create a well-defined Scope, and make sure all team members understand the scope. Documents have requirements written out clearly. Create a Change Management Plan and Communication Plan. |
| RI.5 | Customer patient, social security number, exposed in the form | 3 | Developer | Social Security Number of interviewee published on the form without consent. | Include a consent on the form template the interviewee agrees to before proceeding. |
| RI.6 | Interviewee health information | 2 | Developer | Patient (Interviewee) health information disclosed in form without consent | Include a consent on the form template the interviewee agrees to before proceeding. Also include no warranties “AS-IS,” and UMGC is not responsible for any shortcomings. |

## 3.3 Risk Trigger

A risk register has been created in the table below, showing risk warnings and their ranking number. The risk trigger ranking is used in the Risk Analysis Table 1 as awareness and indication of overall risk.

|  |  |  |
| --- | --- | --- |
| Ranking | Potential Risk | Risk ID |
| 1 | Minor development issue with a requirement that can be resolved by the owner of the risk. | RI.2 |
| 2 | Health Insurance Portability and Accountability Act (HIPPA), patient information disclosed without the patient's consent. | RI.6 |
| 3 | Personally Identifiable Information (PII), the application will record and put into a form patient SSI numbers, mailing or email address, and potential phone number. | RI.5 |
| 4 | Features added that are not approved. | RI.4 |
| 5 | Not able to produce a working application. | RI.1, RI.3 |

# 4. Organization of the project

The mobile application team Project Plan follows the Agile methodology, using the Scrum framework, Section 4.2 will go into more detail of the Agile methodology, and Scrum. Agile Project Management using MS Project will set visual management for the Project manager. Sprints will be held weekly, and if determined more than one should be kept that week, it will be planned. Milestone meetings will be held before the due date of the Milestone. The mobile application team will work with the DevSecOps Team to create an Azure DevOps repository for the Scrum Artifacts.

Because the project follows an Agile methodology with a Scrum approach, there will be two phases with six tasks. The Initiation phase starts with the Project Plan and SRS for Milestone 1 and creates the Product Backlog. The second phase is the Execution Phase, which will have the planned start date, and end date that coincides with Milestone 3. The Execution Phase will consist of: Sprint Planning, Sprint Monitoring, Product Increment, Sprint Review, Sprint Retrospective (Scrum Guides, 2020).

## 4.1 Communication

The primary form of communication will be through MS Teams and email. MS Teams will be used to upload the Milestone documentation so each member can open the Word document and edit the document through Track Changes. This will allow documentation of where the updates or additions were made, who, and when the changes occurred. A conversation can also be started in the document if any team members have a question or comment about the document. If critical documented communications need to be made to the team or, individuals’ email will be used.

Further communication is planned through MS Teams, Mobile Team Channel open chat for all-day meetings. If a video call needs to be placed, it will be determined at the all-day chat meeting. This helps the team members be responsive to questions and answers while addressing different time zones. Reoccurring meetings through MS Teams are scheduled to discuss deliverable to-do lists and address any communication that needs full team support. Milestones will be communicated to all team channels of the project to ensure everyone across project channels know the project scope, development architecture, and communicating effectively to produce a working application.

**Stakeholder Communication Process:** If the Project Stakeholder needs to contact the group, or the team needs to contact the Stakeholder, this will be done through MS Teams or email. A Milestone recap meeting will be routinely held through MS Teams with the Stakeholder. A Project Manager and Stakeholder meeting are held every Sunday, reoccurring through the life of the project.

**Cross Team Communication Process:** Each team has a Developer Liaison/POC that meets weekly as needed to discuss communication topics related to the project; this can be questions, answers, updates, limitations, or changes. The POC will inform the Project Manager and team of the outcomes, and if changes need to occur, it will be documented in a Change Log and emailed to the PM for emergency cases. Otherwise, the Change Log will be checked daily for approval. Project managers will call each other in case of Project emergencies, use MS Teams for daily chat communication as needed, contact one another through email as needed through the week, and a reoccurring Project Manager meeting is scheduled for Sundays. If Dr. Mir Assadullah needs to be contacted, it will be determined at that time.

## 4.2 Explanation of Agile methodology using Scrum framework that will be applied

For the mobile application Project Plan, the method chosen is the Agile Methodology using the Scrum framework. The Product Backlog will be created by the product owner, who will order the task based on a problem. A Scrum Team will then turn the lessons into an increment of value in the Sprint. The Scrum Team and stakeholders perform an analysis of the results to confirm if any adjustments need to be made before moving on to the next Sprint. This process is repeated. No instructions are given to the team members. The team members will follow the rules of Scrum to guide the team interaction. Scrum will allow the team member to work together to share and gain skills. This is done successfully by implementing the Scrum pillars; transparency, inspection, and adaption (Scrum Guides, 2020). A Scrum Team encompasses a wide range of skills that allow the team members to be cross-functional (Scrum Guides, 2020). The members are self-managing in the Sprints allowing them to freely decide which member will complete what, when, and how (Scrum Guides, 2020). Developers will create a plan for the Sprint and Sprint Backlog.

Scrum Events:

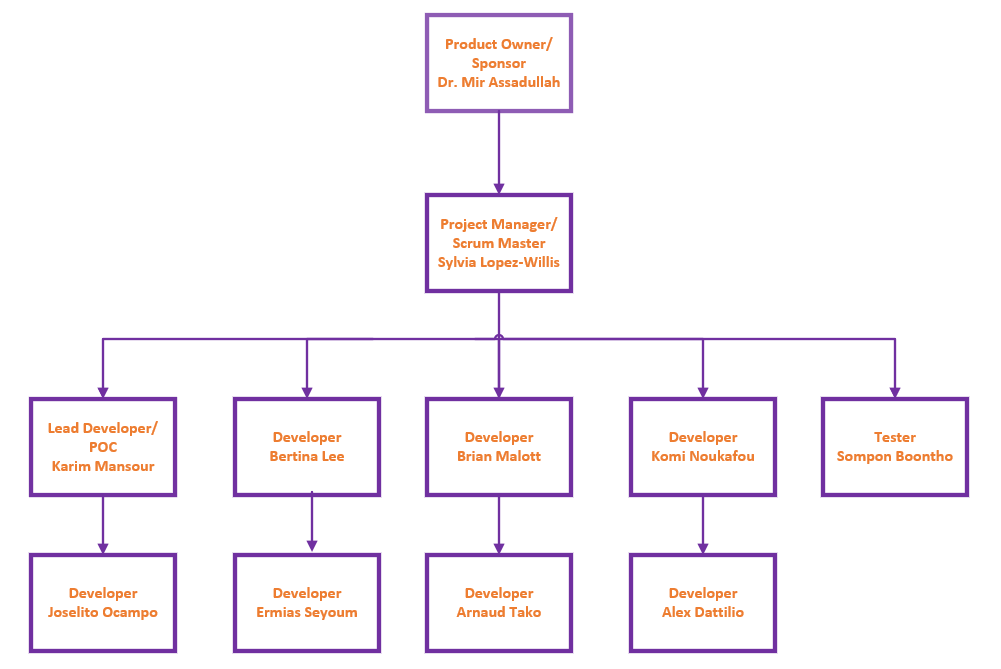
* The Sprint
* Sprint Planning
* Daily Scrum
* Sprint Review
* Sprint Retrospective

Scrum Artifacts:

* Product Backlog
* Sprint Backlog
* Increment

## 4.3 Project Staffing

The team lacks one member that would take on the role of Business Analyst, and no Scrum Master has been assigned. The Project Manager will be the Scrum Master and Business Analyst. Business Analyst tasks will be delegated as needed to other team members if any issues arise. The project will have eight developers that will include a Lead Developer/POC for the team. One tester will take on the BA role, if needed, as a backup.



## 4.4 Roles and Responsibilities

Each team member will be responsible for the role they play and may need to step into other positions depending on the need of the Project.

|  |  |  |
| --- | --- | --- |
| **Role** | **Assignee** | **Responsibility** |
| Project Owner/Sponsor | Dr. Mir Assadullah | As a sponsor for the project Dr. Assadullah, sets the requirements and evaluates the project's advancement. As problems arise, Dr. Assadullah is on standby to give guidance if necessary. |
| Project Manager/BA/Scrum Master | Sylvia Lopez-Willis | The project manager is responsible for monitoring the project's progress, planning the project, coordinating meetings, and ensuring the Agile methodology with a Scrum framework is seen through. Also, watching that the project meets the requirements and no additional features are being added that would take up time or cause the project to go overbudget. |
| Lead Developer POC/Liaison | Karim Mansour | Karim is responsible for leading the team design, development, debugging, and mobile application coding in the Lead Developer role. Karim also is the point of contact for the mobile team with other team Developer POC/liaisons. Delivering the work through Sprints and contributing to the Milestones. |
| Developers | Bertina Lee  Brian Malott  Komi Noukafou  Joselito Ocampo  Ermias Seyoum  Arnaud Tako  Alex Dattilio | Developers are responsible for the coding, designing, development, and debugging of the mobile application. Delivering the work through Sprints and contributing to Milestones. |
| Tester | Sompon Boontho | Review software requirements and set up test scenarios. Analyzing test results and communicating them through the documentation for the developer team. Contributing to Project Milestones. Working as a backup Business Analyst as needed. |

## 4.5 Responsibility Assigned Matrix (RACI)

Because this project is being created in an Agile methodology with Scrum framework, the RACI has been adapted to Scrum.

Responsible = R Accountable = A Consulted = C Informed = I

Facilitator/Coach = F

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RACI Matrix | Sylvia Lopez-Willis | Dr. Mir Assadullah | Karim Mansour | Bertina Lee | Brian Malott | Komi Noukafou | Joselito Ocampo | Ermias Seyoum | Arnaud Tako | Alex Dattilio | Sompon Boontho |  |  |
| Ensure consistency of Scrum practices across teams | C/R/A | C | I | I | I | I | I | I | I | I | I |  |  |
| Provide vision and goal for the product | I | R | I/C | I/C | I/C | I/C | I/C | I/C | I/C | I/C | I/C |  |  |
| Prioritize and manage the product backlog | F/R/A | C | C | C | C | C | C | C | C | C | C |  |  |
| Remove impediments | R | C | R | R | R | R | R | R | R | R | R |  |  |
| Manage the release train | I/R/A | C | R | C | C | C | R | C | C | C | C |  |  |
| Make sure Scrum practices are used and improved within the team | R/A/F | C | R/A/F | R/A | R/A | R/A | R/A/F | R/A | R/A | R/A | I |  |  |
| Create, apply, and continuously improve the definition of done | F | I | R/A | R/A | R/A | R/A | R/A | R/A | R/A | R/A | R/A |  |  |
| Report on time to management | F | C | I | I | I | I | I | I | I | I | I |  |  |
| Define acceptance criteria | F | C | R | C/I | C/I | C/I | R | C/I | C/I | C/I | C/I |  |  |
| Write acceptance tests | F | C | R | R | R | R | R | R | R | R | R/A |  |  |
| Ensure quality of the product | R | C | R | R | R | R | R | R | R | R | R |  |  |
| Manage risks | C/R | C | R | R | R | R | R | R | R | R | R |  |  |
| Approve user stories (user stories meet the acceptance criteria) | F | C | R | R | R | R | R | R | R | R | R |  |  |
| Meet release date and goal | I | C | I | I | I | I | I | I | I | I | I |  |  |
| External SME to review reports | R | C | R | R | R | R | R | R | R | R | R |  |  |

## 4.6 Project Tools

The table below shows the tools that the Project will use to produce the mobile application.

|  |  |
| --- | --- |
| Tool | Function |
| Software Engineering Project, SWEN 670 Online Course, LEO | Through LEO will be the location where the delivery of the Milestones and finished product will be uploaded. |
| Google Play Store | The Android mobile application will be published to the Google Play Store. |
| App Store | The iOS mobile application will be published on the iOS App Store. |
| Android Studio | The Android Studio integrated development environment will be used to develop, edit, and compile Flutter code. |
| Flutter | A Google UI toolkit for developing natively compiled mobile applications with a single codebase. |
| Dart | An object-oriented programming language will be used to create the mobile application. |
| Github | The Github repository will organize the project for code version control and collaboration between team members |

## 4.7 Project Document Storage

The Milestones 1 through 4 will be stored in MS Teams, using SharePoint. A Github repository was created for the mobile team to store the mobile application project work. This can be accessed remotely from anywhere in the world. This promotes a collaborative environment for remote work.

## 4.8 Deliverables

This project's deliverables can be viewed in detail in [Appendix B](#_Appendix_B). Milestone 1 consists of a Project Plan and a Software Requirements Specification document. Milestone 2 will be the Technical Design Document, a Software Test Plan will be added to the Project Plan, and if any updates need to be made from Milestone 1, it will be submitted in Milestone 2. Milestone 3 includes a Programmer Guide, Deployment and Operations Guide (Runbook), and a draft of the Research Paper. Milestone 4 consists of a User Guide, Test Report, and the final research paper.

## 4.9 Initiation phase

The Initiation Phase of the project is the project plan. The SWEN 670 Kick-Off Meeting was first held with the stakeholders and graduate students taking the course. Roles were approved by skill and experience, and then teams were created. The Product Backlog is started and reviewed in this phase, such as work done, activities will be in line to produce a project schedule.

Please refer to [Appendix A](#AppendixA) for a Work Breakdown Structure (WBS) of the Initiation Phase.

## 4.10 Execution Phase

Planned starting date: 2/10/2021

Planned end date: 3/23/2021

Description of the Execution Phase:

During the Execution Phase, five tasks are executed that are part of the Seven-phase Scrum approach. Since this project follows an Agile methodology with a Scrum approach, the project schedule has been adapted to reflect this.

The Sprint Planning task will involve all the developers and Scrum Master to discuss the Sprint Goals and then divide those goals into a Sprint Backlog.

Sprint Monitoring task will be tracked by the Scrum Master and Lead Developer, with the Sprint Backlog. If, after analysis, it is determined there are any risks to the project, issues will be removed by the PM/Scrum Master and Lead Developer.

A Scrum Meeting will be held daily through the MS Teams, Mobile Team Channel chat since graduate students are located in different time zones and work during the day.

Product Increment is the task that will require any changes to the Backlog Requirements to be done through analysis, development, testing, and then documented. Test Cases and Test Results will be documented and created during this task.

The project team will analyze if the product is functioning correctly based on the Stakeholder's requirements in the Sprint Review task. If it is determined that the product is not functioning as expected, the areas to be addressed will be put back in the Product Backlog as priority. The team will then discuss with the Project Manager/Scrum Master the status of the analysis. The Product Backlog and Sprint Backlog are reviewed.

During a Sprint Retrospective, the team will review lessons learned and make the next Sprint or work better.

### 4.10.1 Change Management Plan

Change control will have a set guideline to follow, so all teams communicate with one another. A process has been put in place to require notifications and approval from the primary Stakeholder, Dr. Mir Assadullah. The Project Managers across all teams will meet to discuss the change request and present it to the overall Project Manager that forms the Change Control Board. The overall Project manager will present the change to Dr. Mir Assadullah for approval.

Email communication will be sent to the Project Manager and the Mobile Team members of any changes, and a meeting will be held to discuss the said change request. An evaluation will be done based on the severity of the change. A Change Log will be used and available through MS Teams, Mobile Team Channel to track changes. The Mobile Team will communicate the change request with other team channels to stay in communication, authorize, and agree to the change.

If a change has been identified as an emergency, the Project Managers have agreed to call one another through a group call the same day of the change notification. This will allow the project to stay on course and not get off track.

# 5 Cost and Schedule Overview

The final section of the project plan provides an overview of the costs and the timeline for the entire project.

## 5.1 Cost Overview

The table below shows a timeline and the cost of the whole project. Since the Initiation and Execution phase are both worked into the Milestones, those section costs have been left blank. The price was estimated on each student role based on hourly pay and a $300 credit from Google.

Table 5 - Overview of Costs

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Expected starting date** | **Expected date of completion** | **Total cost estimate** |
| **Milestone 1** | **1/13/2021** | **2/2/2021** | **$11921.79** |
| **Milestone 2** | **2/3/2021** | **2/23/2021** | **$11921.79** |
| **Milestone 3** | **2/24/2021** | **3/16/2021** | **$11921.79** |
| **Milestone 4** | **3/17/2021** | **3/30/2021** | **$7959.86** |
| **Initiation** | **1/13/2021** | **2/9/2021** |  |
| **Execution** | **2/10/2021** | **3/23/2021** |  |
|  |  | **Total:** | **$43,725.23** |

## 5.2 Schedule Overview

The schedule in [Appendix A](#_Appendix_A_-) shows a complete breakdown of the overall schedule. The schedule begins with gathering requirements and communication with the team. Milestone 1, 2, 3, and 4 are broken down into deliverables. Followed by the Milestones schedule is the Initiation phase that is given four weeks to complete and is described in more detail in [Section 4.9 Initiation Phase](#_4.9_Initiation_phase). The Execution Phase – Scrum lasts six weeks, and this is where the development starts. A more in-depth description can be referenced in [Section 4.10 Execution Phase](#_4.10_Execution_Phase).

# 6 Software Test Plan

## 6.1 Test Plan Identifier

Master Software Test Plan for Mobile Application (STPMA).

## 6.2 Introduction

## 

The mobile team will design and develop an artificial intelligence (AI) mobile application by using the Flutter and Android studio platform. The application will include the following use cases:

1. Sliding menu
2. Log In
3. Log Out
4. Begin Conversation Session
5. Specify a From Template to Form Scriber
6. Specify “New Prescription” to Form Scriber
7. Specify “Medical History” to Form Scriber
8. Request Report
9. View Reports
10. View Settings

11. View Help.

The software test will cover all functionalities, including the customer interface, to verify and validate that our mobile application works well and accomplish their tasks. The primary tasks of this software test will focus on the following categories:

1. Unit tests
2. Widgets tests
3. Integration tests
4. Continuous integration services.

### 6.2.1 Background

The Software Test Plan for Mobile Application (STPMA) has been created to test the UMGC AI mobile application to assure that the software system meets the software requirements specification, the user, and stakeholders expectations. The STPMA serves as the master test plan for the mobile application. The tests will be performed throughout the project life cycle. The test results will be reported to the project manager (PM), the developers, and the stakeholders. Negative test results and bugs will be fixed to improve the quality and performance of the software system.

### 6.2.2 Objective of the Test Plan

Software testing is a method to check whether the actual software product matches expected requirements and ensure that it is defective. It involves the execution of software system components using manual or automated tools to evaluate one or more interest properties. The purpose of software testing is to identify errors, gaps, or missing requirements in the product.

This software test plan aims to communicate to the development team and stakeholders a detailed plan for running development tests. The Software Test Plan is used to validate end-to-end business flow. It is a kind of black-box testing where two or more end-users will be involved.

### 6.2.3 Objective of STPMA

The STPMA objective for the mobile application will ensure that the stakeholder requirements are met, and the final product will be delivered to public service professionals with functional features. The STPMA will follow the Software Requirements Specification when approaching the software testing. Specific attention will be placed on the Use Cases section of the SRS for the STPMA.

## 6.3 Reference Documents

* Lee, Bertina; Lopez, Sylvia; Malott, Brian; Mansour, Karim; Noukafou, Komi., 2021: *Software Requirements Specification document for the Mobile Team*
* IEEE29 International Testing Standards
* Lopez\_Willis, S., 2021: *Project Plan Mobile Team*.

## 6.4 Test Items

Before launching the Form Scriber mobile application, eleven Use Cases from the SRS will be tested by the tester and development team to validate and verify the mobile application.

Table 1 - test item is based on Use Case: Sliding Menu, the user swipes right to left to see the sliding menu options.

Table 1 – Sliding Menu

|  |  |  |  |
| --- | --- | --- | --- |
| **Items to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 1. Sliding menu | 1.0 | 1. The actor swipes right to left | 2. The system displays the menu to allow the user to click on the menu item for the Homepage, Begin Conversation, View Reports, Settings, Help and Log Out |
|  |  |

Table 2 test item is based on Use Case: Login, the display and functionality of logging a user into the system.

Table 2 - Login

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 2. Login | 1.0 | 1. The actor taps on the login button on the landing screen. | The system will display the login screen with the Username and Password fields. |
| 2. 3. The actor enters their Google credentials, username and password, then taps the login button | The system will validate the credentials. |
|  |  |  | The system displays the main page. |

Table 2.1 Is continued from the test item shown in Table 2 Login.

Alternate paths:

* If the user enters an invalid username and/or password.

Table 2.1 – Continuation of Table 2 Login, Test Item with Alternative Path

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 2. Login | 1.0 | 1. The user enters an invalid username and/or password. | The system displays an error message indicating the username and/or password is invalid, the system remains on the login screen. |

Table 3 test item is based on the Use Case: Logout, the display and functionality of logging a user out of the system.

Table 3 - Logout

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 3. logout | 1.0 | 1. The actor taps on the logout button on any screen. | The system will end the session and displays a logout confirmation screen. |

Table 4 test item is based on the Use Case: Begin Conversation Session. Navigate to the “Form Scriber” chatbot activity to begin the conversation session. Connects to Form Scriber to initiate communication.

Table 4 - Begin Conversation Session

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 4. Begin conversation session. | 1.0 | 1. The actor taps the Begin Conversation button on the main screen | The system requests a list of available form templates through the Google Drive API, based on the Google Drive repository which the customer has preconfigured. This list populates a DropdownButton menu. The system navigates to the conversation screen and prompts the user to select a template from the drop down list before they can begin speaking. |
|  |  | 2. The actor taps on the drop-down menu. | The system reveals a list of available form templates. |

Table 5 test item is based on the Use Case: Specify “Patient Registration” to Form Scriber. Sends formName for “Patient Registration” to Form Scriber.

Table 5 - Specify a Form Template to Form Scriber

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 5. Specify a Form Template to Form Scriber. | 1.0 | 1.The actor selects the form template from the drop down list. | 1.The system displays the form name on screen and gets the URL in memory, stored as a String. |
|  |
|  |  | 2. The actor clicks “Proceed”. | 1.  The system sends the form URL corresponding to the selected form name to the DialogFlow agent and receives a response from DialogFlow indicating it’s ready to listen for field input. |
|  |  | 3. The actor taps the microphone button. | 1. The system stops recording the audio. |
|  |  | 4. The actor speaks into their microphone, stating the name of the specific field they want to update. | 1.The system automatically  stops recording audio after a standard moment of silence of 2 seconds, and converts their audio into text and sends it to the DialogFlow agent. |
|  |  |  | 2. The DialogFlow agent responds with a confirmation message that the desired intent is ready to be recorded. |
|  |  | 5. The actor taps the microphone button and speaks the field entry detail to get written. | 1. The system converts audio and sends this message as a test to the DialogFlow agent, then receives a response from DialogFlow confirming the specific field was written. |

Table 6 test item is based on the Use Case: Specify “New Prescription” to Form Scriber. Sends formName for “New Prescription” to Form Scriber.

Table 6 - Specify "New Prescription" to Form Scriber

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 6. Specify “New Prescription” to Form Scriber | 1.0 | 1. The actor selects the form from the From Bot chatbot suggestion. | The system sends the formName corresponding to New Prescription to the DialogFlow agent and receives a response from DialogFlow. |
|  |
|  |  | 1. The actor taps the microphone button | The system starts recording the audio. |
|  |  | 1. The actor taps the microphone button | The system stops recording the audio. |

Table 7 test item is based on the Use Case: Specify “Medical History” to Form Scriber, sends form ID for “Medical History” to Form Scriber.

Table 7 - Specify "Medical History" to Form Scriber

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 7. Specify “Medical History” to Form Scriber. | 1.0 | 1. The actor selects the form from the Form Scriber chatbot suggestion. | The system sends the formName corresponding to New Prescription to the DialogFlow agent and receives a response from DialogFlow. |
|  |
|  |
|  |
|  |  | 1. The actor taps the microphone button | The system starts recording the audio. |
|  |  | 1. The actor taps the microphone button | The system stops recording the audio. |

Table 8 test item is based on the Use Case: Request Report, sends form URL of the selected template to Form Scriber.

Table 8 - Request Report

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 8. Request Report | 1.0 | 1. The user provides the required data in the conversation in the chatbot.  2. The user will indication the request for completed report by saying “print”. | The system will request the report URL from the DialogFlow agent. |
| The system will receive the report URL from DialogFlow. |

Table 9 test item is based on the Use Case: View Reports, displays report history.

Table 9 - View Reports

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 9. View Reports | 1.0 | 1. The actor taps on the View Reports link from the menu navigation. | The system requests a list of links to the customer’s designated Google Drive document repository. The system displays the list of reports available to view. |

Table 10 test item is based on the Use Case: View Settings, displays and sets settings.

Table 10 - View Settings

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 10. View Setting | 1.0 | 1. The actor taps the Settings link in the menu navigation. | 1. The system displays the available settings on the Settings screen. |

Table 11 test item is based on the Use Case: View Help, displays application’s help system.

Table 11 - View Help

|  |  |  |  |
| --- | --- | --- | --- |
| **Item to be Tested** | **Version Number** | **Test Action** | **Expected Results** |
| 11. View Help | 1.0 | 1. The actor taps the Help link on a screen in the system. | 1. The system displays the help screen. |

## 6.5 Feature to be tested

Features that will be tested for the Form Scriber mobile application are listed in the bullet points below.

* Secure login to the system
* System responds to the actor with help menu available
* Easy logout from the system
* User friendly for a conversation session
* User receives form URL of the selected template
* Easy to enter New Prescription to Form Scriber
* Providing secure Medical History to Form Scriber
* Providing secure Request Report
* Easy to view reports
* Easy to view settings
* Providing useful help menu.

## 6.6 Features not to be tested

Features that are not included in the requirements will be out of scope and not tested for the Form Scriber mobile application. Any requirements not listed in Section 6.5 will not be tested.

## 6.7 Approach

An approach to testing will be broken down into five sections. Those processes are listed below.

1. Develop test
2. Prepare the test
3. Run the test
4. Review the test
5. Report the test.

### 6.7.1 Develop Tests

The test tools below listed by the bullet points will be used to develop tests.

* Firebase Test Lab
* Flutter test function.
* Android Studio

Flutter test can be used to:

* Perform Unit tests to validate the add and remove operations
* Perform Widgets tests for the home and favorites pages
* Perform UI and performance tests for the entire app using Flutter Driver.

Android Studio is designed to make testing simple. With just a few clicks, we can set up a JUnit test that runs on the local JVM or an instrumented test that runs on a device. We can also extend our test capabilities by integrating test frameworks such as [Mockito](https://github.com/mockito/mockito) to test Android API calls in our local unit tests and [Espresso](https://developer.android.com/topic/libraries/testing-support-library#Espresso) or [UI Automator](https://developer.android.com/topic/libraries/testing-support-library#UIAutomator) to exercise user interaction in the instrumented tests. We can generate Espresso tests automatically using [Espresso Test Recorder](https://developer.android.com/studio/test/espresso-test-recorder).

The mobile team will perform the following test types during System Integration Testing:

1. Acceptance testing: This test will be performed during the system integrating phase.
2. Unit testing: This test will be performed during the system development phase.
3. Functional testing: This test will be performed during the use case development cycle.
4. Regression testing: This test will be performed throughout the SDLC.

Test development includes the following activities:

1. Reviewing and analyzing the Project Plan and the Software Requirements Specification.
2. Development of scenarios and required testing techniques,
3. Define acceptance criteria to meet the accepted standard.
4. Construction of test case which is defined based on the use case input against the expected output of each use case.
5. Writing and creation of test scripts.
6. Running the tests and documents the test result of each test.
7. Review of all testing documentation for that test.

The following reviews will be conducted by the tester, the project manager, the development team leader, and the developers.

1. Test plan review
2. Test case review
3. Test progress and milestone reviews
4. Post-test review.

### 6.7.2 Prepare to Test

A preparation to test the Form Scriber mobile application will have the following processes:

1. Preparation of test environment: hardware, testing tools, and staffing will be functional to allow the testing to be rolled out.
2. Preparation of testing documentation: test documentation will be created for setting test guidelines and reports.

### 6.7.3 Run Tests

We can use the following processes to run the test.

1. Test scripts and test cases will be generated in the appropriate assigned development section of the mobile application.
2. Tracking tests done through one of the testing tools listed in Section 6.7.1 Develop Tests.

### 6.7.4 Review Test Results

Evaluation will be done for each component under the test, and results will be documented. Deviations will be addressed with valid reasons and possible action items to correct them in future releases. Test results will be checked against requirements and system impact. The developer for the assigned development task will work with the tester to review the results.

Defects will have ratings determined by severity and priority that will be identified as critical, major, moderate, or low. Table 15 shows a chart with the first column showing Defect Priority Levels and the second column displaying Defect Severity Levels.

Table 15 - Defect Rating Chart



### 6.7.5 Test Results and Defects

Severity defines the extent to which a particular defect could create an impact on the application or system.

Severity is a parameter to denote the implication of a defect on the system – how critical the defect is and its impact on the whole system’s functionality. The severity is a parameter set by the tester while he opens a defect and is mainly in control of the tester.

**Broadly the defects can be classified as follows:**

#### **Critical (S1)**

A defect that ultimately hampers or blocks testing of the product/ feature is a critical defect. An example would be in the case of UI testing, where after going through a wizard, the UI hangs in one pane or doesn’t go further to trigger the function. Or in some other cases, when the feature developed is missing from the build.

For any reason, if the application crashes or becomes unusable or unable to proceed further, the defect could be classified under critical severity.

Any catastrophic system failures could lead the user to non-usability of the applications could be classified under the Critical severity

#### **Major (S2)**

Any Major feature implemented that is not meeting its requirements or use case(s) and behaves differently than expected, it can be classified under Major Severity.

A major defect occurs when the functionality is functioning grossly away from the expectations or not doing what it should be doing. Any defect that could lead to incorrect data persistence, data issues, or wrong application behaviors could be broadly classified under the Major severity.

**Minor/Moderate (S3)**

Any feature implemented that is not meeting its requirements/use case(s) and behaves differently than expected the impact is negligible to some extent, or it doesn’t have a significant effect on the application, can be classified under Minor Severity.

A moderate defect occurs when the product or application doesn’t meet specific criteria or exhibits unnatural behavior. However, the functionality as a whole is not impacted.

#### **Low (S4)**

Any cosmetic defects, including spelling mistakes or alignment issues, or font casing, can be classified under Low Severity.

A minor low severity bug occurs when there is almost no impact on the functionality, but it is still a valid defect that should be corrected. Examples of this could include spelling mistakes in error messages printed to users or defects to enhance the look and feel of a feature.

**Broadly, Priority of the defects can be classified as follows:**

#### **Priority # 1 Critical (P1)**

This has to be fixed immediately within 24 hours. This generally occurs when an entire functionality is blocked, and no testing can proceed due to this. Or in certain other cases, if there are significant memory leaks, then generally, the defect is classified as a priority P1 meaning the program or feature is unusable in the current state.

Any defect that needs immediate attention which impacts the testing process will be classified under the immediate category

All the **Critical severity**defects fall under this category (unless re-prioritized by business/stakeholders)

#### **Priority #2 High (P2)**

Once the critical defects have been fixed, a defect having this priority is the next candidate, which has to be fixed for any test activity to match the “exit” criteria. Usually, when a feature is not usable as it’s supposed to be due to a program defect, or that new code has to be written, or sometimes even because some environmental problem has to be handled through the code, a defect may qualify for a priority 2.

This is the defect or issue which should be resolved before the release is made. These defects should be resolved once the Critical issues are solved.

All the **Major** **severity** defects fall into this category.

#### **Priority #3 Medium (P3)**

A defect with this priority must be in contention to be fixed as it could also deal with functionality issues that are not as per expectation. Sometimes even cosmetic errors such as expecting the right error message during the failure could qualify to be a priority three defect.

This defect should be resolved after all the serious bugs are fixed.

Once the Critical and the High priority bugs are done, we can go for the medium priority bugs.

All the**Minor** **severity** defects fall into this category.

#### **Priority #4) Low (P4)**

A defect with low priority indicates an issue, but it doesn’t have to be fixed to match the “exit” criteria. However, this must be resolved before the GA is done. Typically, some typing errors or even cosmetic errors, as discussed previously, could be categorized here.

Sometimes defects with priority low are also opened to suggest some enhancements in the existing design or a request to implement a small feature to enhance user experience.

This defect can be resolved in the future and does not need any immediate attention, and the **Low severity** defects fall into this category. As already discussed, priority determines how quickly the defect turnaround time must be. If there are multiple defects, the priority decides which defect has to be fixed and verified immediately versus which defect can be fixed a bit later.

## 6.8 Item Pass/Fail Criteria

## 

Test cases will be created by the tester for each requirement listed in, [Section 6.5 Feature to be Tested](file:///C:\Users\lysa1117\AppData\Local\Microsoft\Windows\INetCache\Content.Outlook\DZMP9OWZ\Updates_Project%20Plan%20Milestone%201_Mobile%20Team.docx#_6.5_Feature_to). Test cases will include user inputs and the appropriate output. The test will pass based on the test case outcome and the predefined output. To determined if a test has failed, the test output will be reviewed, and if it does not match the predefined outcome, it will be documented as failed.

## 6.9 Test Plan Matrix

A matrix will be put in place for the mobile application to record tests, document test coverage, what tests were done, where the defect might be, and assure requirements were met.

Table 16 shows the Test Plan Matrix, with each column labeled. The first column will list Test Cases Planned, the second column Test Cases Executed, the third column Test Cases Passed, and the last column Test Cases Failed. What has been entered into Table 16 is an example of what kind of data will be inputted.

Table 16 – Test Plan Matrix

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Cases Planned** | **Test Cases Executed** | **Test Cases Passed** | **Test Cases Failed** |
| Login | 1. The actor taps on the login button on the landing screen. | 80% | 20% |
|  |  |  |  |

## 6.10 Test Deliverables

The mobile team will deliver to the tester the outcome of a test:

* Testing Acceptance Plan, which describes overall activities of the mobile application tests.
* The results of each test case.
* Test log.
* Incident report.
* Incident report log update.
* Test Summary Report.

## 6.11 Testing Tasks

Testing tasks will be adapted in the following manner:

1. Software Test Plan
2. Communication with the Lead Developer of the assigned requirements for the test deliverables following corresponding documentation.
3. Confirm that proper testing tools are being used and developers are trained to know how to use them.
4. Run tests.
5. Tests are documented.

## 6.12 Environmental Needs

The following tools will be needed to process the testing that will be done for the mobile application:

* Flutter test
* Android Studio
* Firebase Test Lab

## 6.13 Responsibilities

The Lead Tester will be responsible for evaluating the requirements of the mobile application attributes and capabilities. Then organizing and managing the testing process to verify that visibility, traceability, and control of tests are met to deliver a working mobile application.

For the testing to be successful, the Lead Tester and Lead Developer will work together with other team channels to ensure requirements are met. Table 16 shows the assigned test tasks and which team member will be responsible for the specified task.

Table 17 - Test Responsibilities

|  |  |
| --- | --- |
| Task | Assigned To |
| Software Test Plan | Sompon Boontho, Sylvia Lopez-Willis |
| Requirements Documentation | Sompon Boontho, Sylvia Lopez-Willis |
| Test Creation | Sompon Boontho, Arnaud Tako, Karim Mansour, Bertina Lee, Brian Malott, Komi Noukafou, Joselito Ocampo, Ermias Seyoum, Alex Dattilio |
| Run Test | Sompon Boontho, Arnaud Tako, Karim Mansour, Bertina Lee, Brian Malott, Komi Noukafou, Joselito Ocampo, Ermias Seyoum, Alex Dattilio |
| Summary of Test | Sompon Boontho, Arnaud Tako |

## 6.14 Staffing and Training Needs

Testing will be performed by the tester, developers, and lead developer of the mobile team. The Project Manager and Lead developer will work together to ensure the requirements are met in conjunction with the test results.

The following bullet points will address that the tests are run successfully:

* The mobile team will meet regularly during sprints to ensure testing tools are used and that the tools work.
* Through MS Teams weekly meeting screen sharing, the team will be trained on the testing tool. A video of the session will be posted in the MS Team mobile channel to reference it when needed.
* The mobile team will be provided with the Software Test Plan to review.
* Lead Tester, Lead Developer will work together to communicate tests to developers.

## 6.15 Schedule

Regarding the schedule, Sprints will be held weekly to include Product Increment (Test Cases and Test Results) that will be held throughout the Sprint. Sprint 1 will start Wednesday, February 10, 2021, and the last Sprint 6 will end March 23, 2021. With reference to the Appendix A of the Project Plan, the testing and Sprint schedule can be seen in more detail.

Table 16 shows the Staffing and Training Needs Table, with the first column showing the Task Name, second column Duration, third column Start, fourth column Finish, and fifth column Resource Name.

Table 18 - Staffing and Training Needs Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Task Name** | **Duration** | **Start** | **Finish** | **Resource Name** |
| Software Test Plan | 2 wks | Mon 2/8/21 | Fri 2/20/21 | Sompon Boontho, Sylvia Lopez-Willis |
| Performing Test Cases | 4 wks | Mon 2/23/2021 | Tue 3/23/2021 | Sompon Boontho, Arnaud Tako |
| Test Report | 3 days | Wed 3/24/2021 | Fri 3/30/2021 | Sompon Boontho, Arnaud Tako |

## 6.16 Risks and Contingencies

Each risk and its cause should be identified, the likelihood of it happening should be addressed. A risk management plan should be created. The risk management process needs a schedule to determine how often and when risk activities should occur throughout the project. Qualitative risk analysis qualifies the risks that have been identified in the project. Not all risks are worth responding to, but some demand attention. Qualitative analysis is a subjective approach to organizing and prioritizing risks. Identified risks can be rated according to probability and potential impact. See table 18 for an example:

Table 18 – Risks and Contingencies

|  |  |  |  |
| --- | --- | --- | --- |
| **Risks** | **Probability** | **Impact** | **Risk Score** |
| Mobile app project cannot complete on time. | Low | High | Moderate |
| Use cases development completed late | Moderate | Moderate | High |
| Testing delay due to  Use case late deliverables. | Low | Low | Moderate |
| Project integration with Dialog Flow encounters a technical problem | Moderate | High | High |

The high score risks require more attention to be mitigated and have more priority than the others.

## 6.17 Test Revision History

A table has been created to track test changes. The test changes log will be available in MS Teams mobile channel file folder labeled “Test Revision History Log.” Table 18 shows an example of the log.

Table 19 - Test Revision History Log

|  |  |  |  |
| --- | --- | --- | --- |
| Test Name | Date | Description | Approved By |
|  |  |  | Sompon Boontho |
|  |  |  |  |

# References

Baars, Wouter; Harmsen, Hank; Kramer, Rutger; Sesink, Laurents; Zundert, Joris van. (2006). Appendix 10: Sample project plan. In *Project Management Handbook* (pp. 1 - 8). San Francisco.

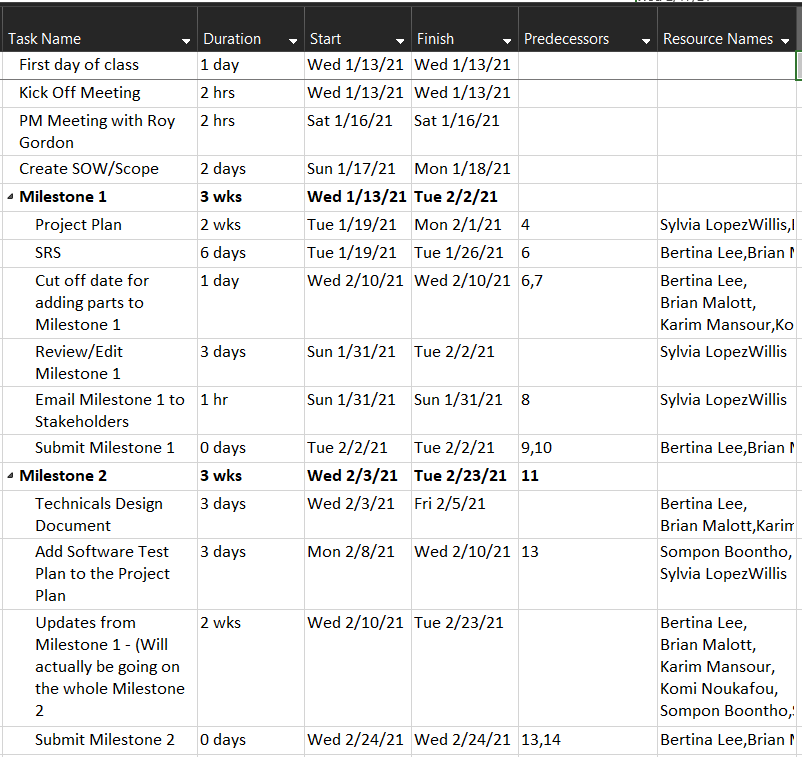
Project-Managment.com. (2019, November 14). *How to Write SMART Project Management Goals*. Retrieved from Project Management: https://project-management.com/what-are-smart-objectives-and-how-to-write-them/

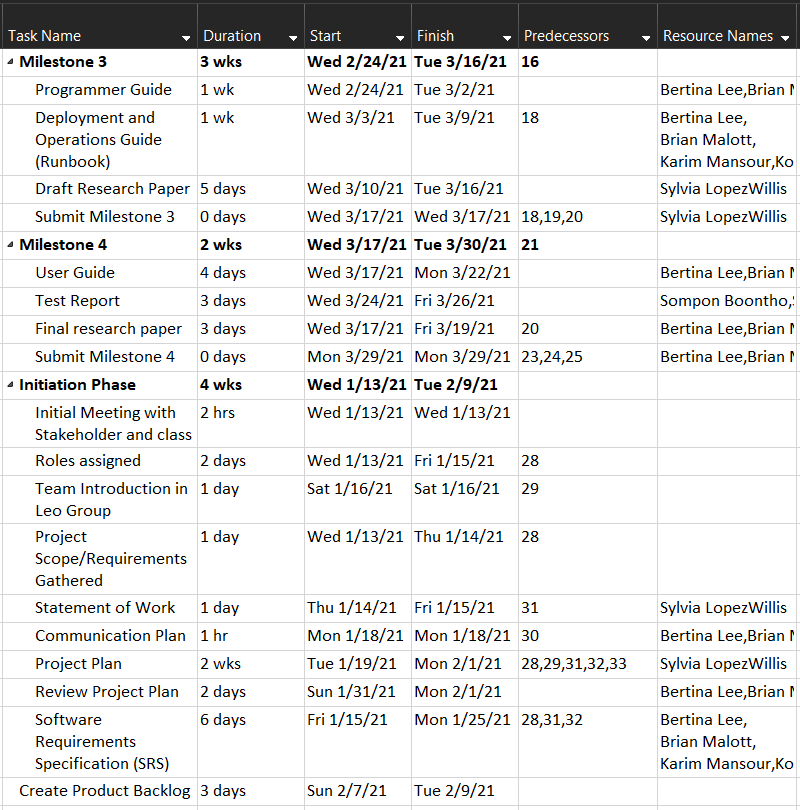
Ray, Stephanie. (2017, October 19). *The Risk Management Process in Project Management*. Retrieved from Project Manager: https://www.projectmanager.com/blog/risk-management-process-steps

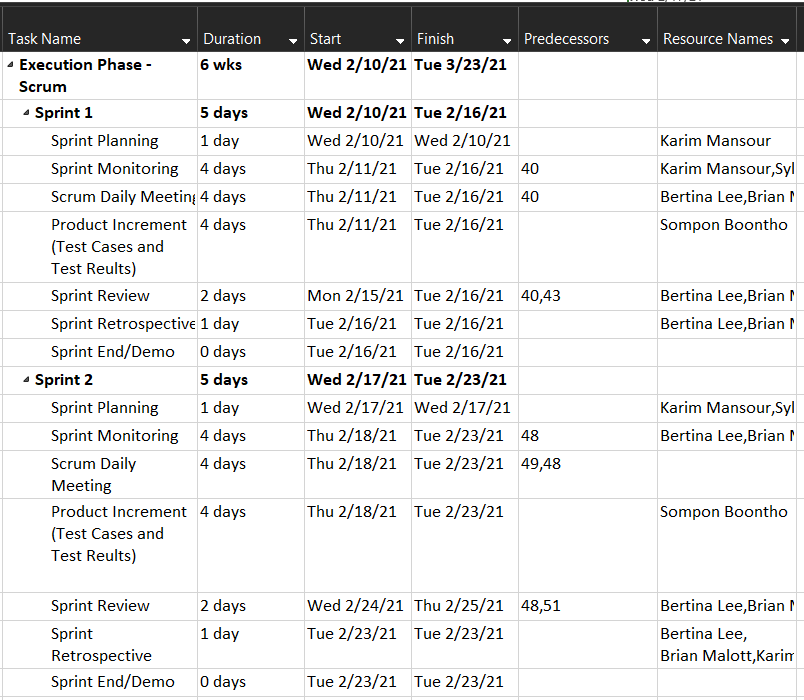
Scrum Guides. (2020). *The 2020 Scrum Guide*. Retrieved from Scrum Guides: https://www.scrumguides.org/scrum-guide.html#increment

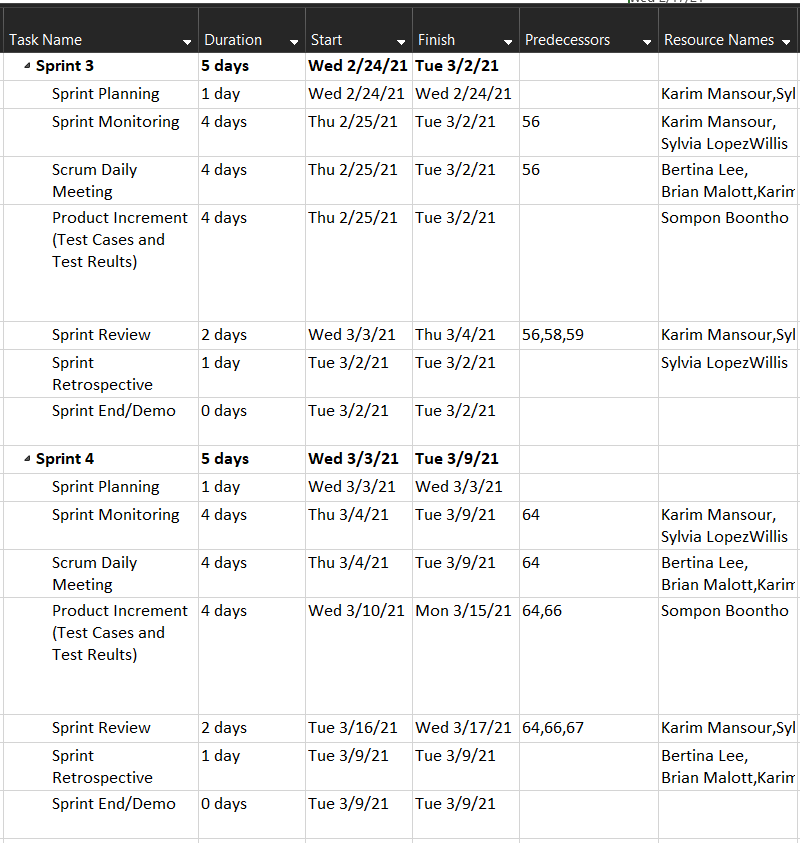
# Appendices:

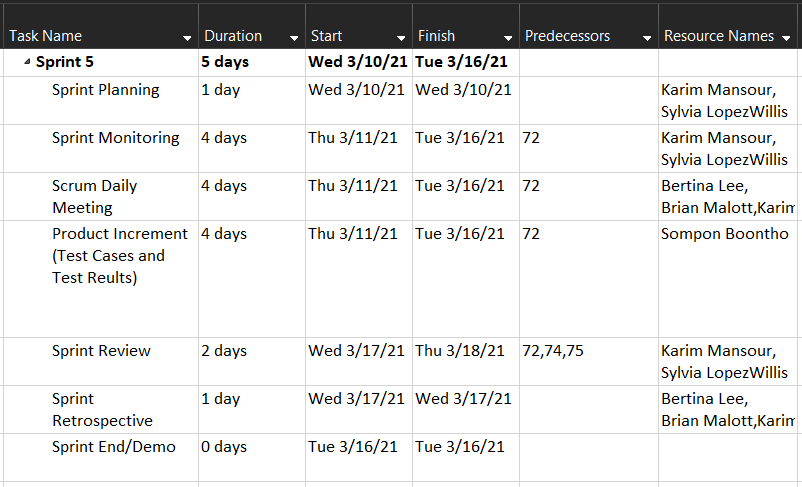
## Appendix A - WBS











## Appendix B – Schedule of Deliverables

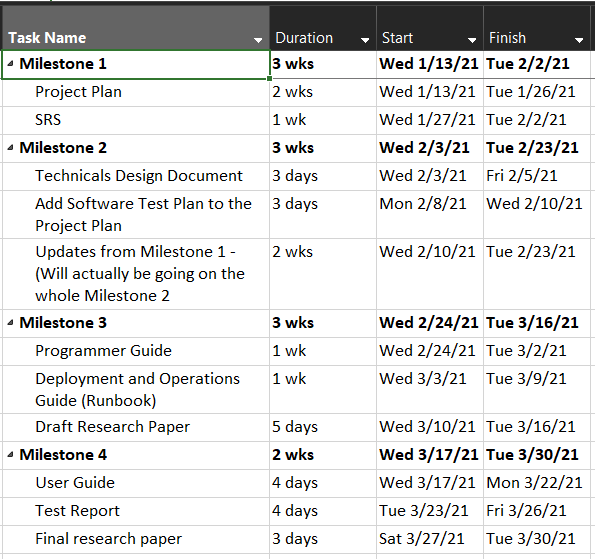


Figure 1 Schedule of Deliverables

## Appendix C – Project Timeline

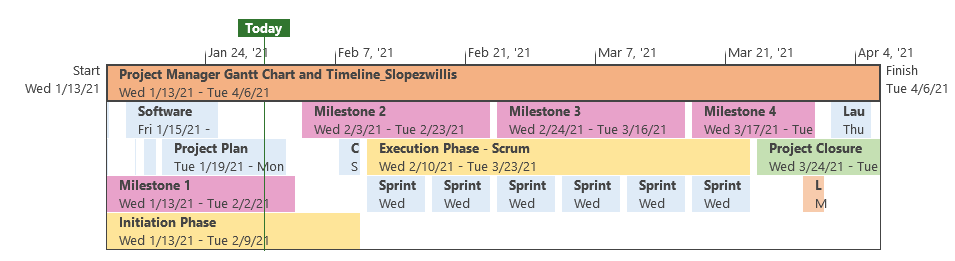


Figure 2 - Project Timeline