### Software Project Management Plan (SPMP)

### 

### Mobile Group Application

### 

### Version 1.0

### 

### January 25, 2019

### 

### https://canvas.biola.edu/courses/28527/assignments/284755

### 

### Department of Computer Science, Biola University

# Revisions

### Overview

Create a software application that is meant to organize groups within Biola University, especially Student Programming & Activities (SPA) clubs, as well as to foster communication and provide information in a manner that is both simple and efficient

### Target Audience

Students at Biola University

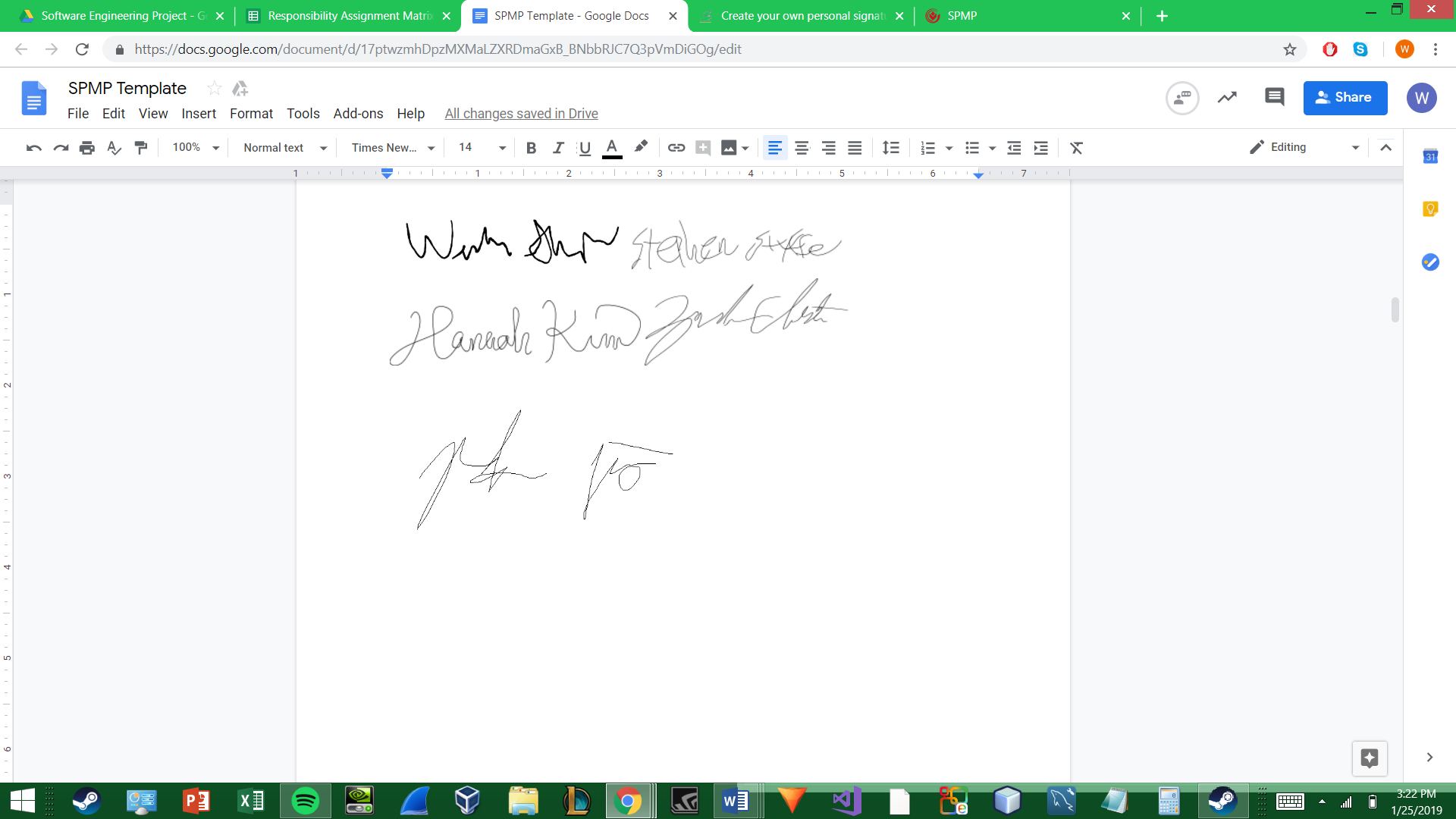
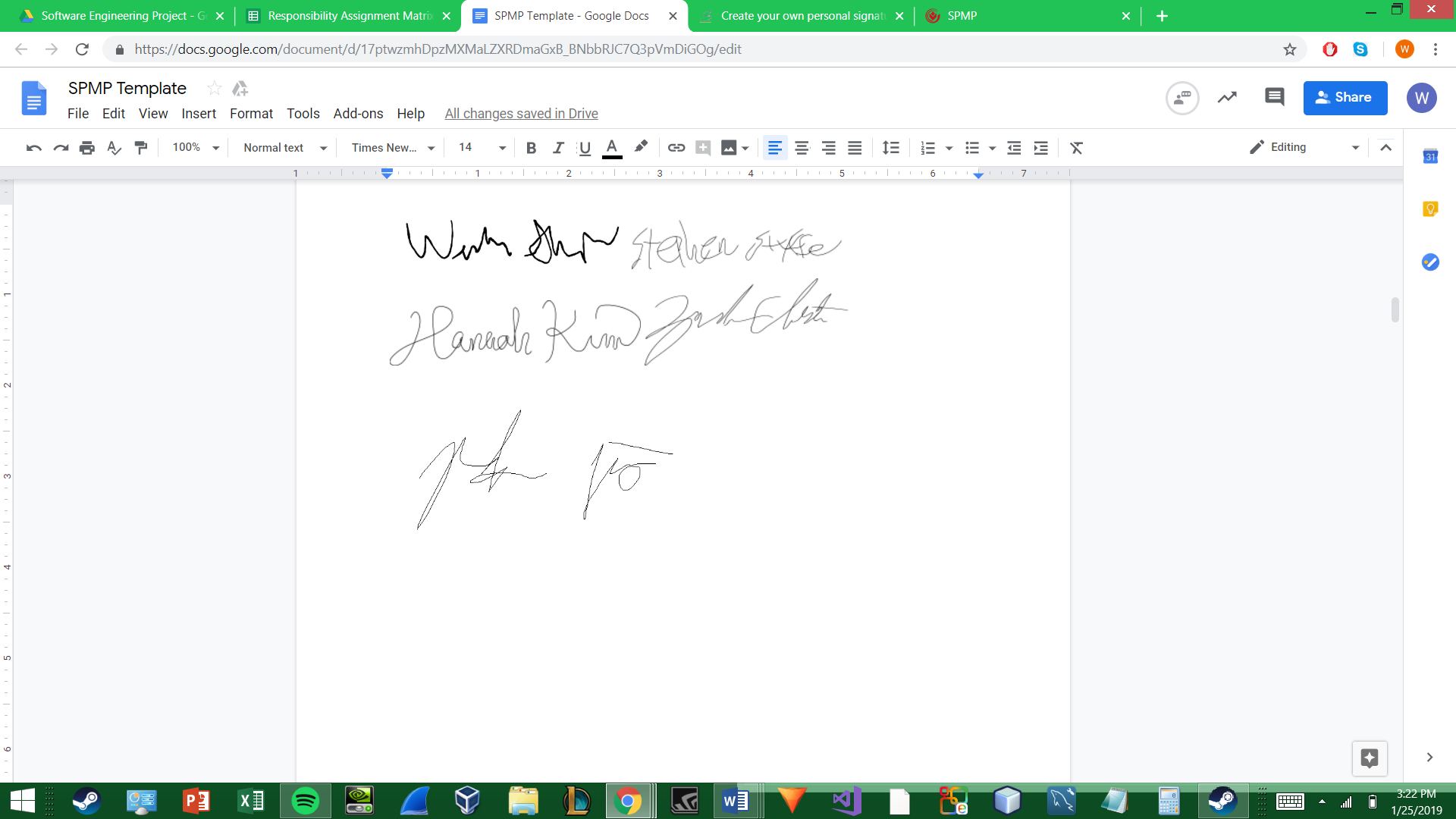
### Project Team Members

Zachary Chester, Jiahui Jiang, Hannah Kim, Wesley Shiozaki, and Stephen Styffe

### Version Control History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Primary Author(s)** | **Description of Version** | **Date Completed** |
| 1.0 | Project Team | Brainstorming and determining basic overall information and ideas | 1/25/2019 |
|  |  |  |  |

### Signatures of Approval



# Table of Contents

1. **Introduction**
   1. Project Overview
   2. Project Deliverables
2. **Project Organization**
   1. Software Process Model
   2. Roles and Responsibilities
   3. Tools and Techniques
3. **Project Management Plan**
   1. Tasks
      1. Task 1 – Setting Up Android Studio Project
         1. Description
         2. Deliverables and Milestones
         3. Resources needed
         4. Dependencies and Constraints
         5. Risks and Contingencies
      2. Task 2 – Create Proxy Server on gCloud Instance
         1. Description
         2. Deliverables and Milestones
         3. Resources needed
         4. Dependencies and Constraints
         5. Risks and Contingencies
      3. Task 3 – Design MySQL Database
         1. Description
         2. Deliverables and Milestones
         3. Resources needed
         4. Dependencies and Constraints
         5. Risks and Contingencies
   2. Assignments
   3. Timetable
4. **Additional Material**

# Introduction

### Project Overview

To create a software application that is meant to organize groups within Biola University, especially Student Programming & Activities (SPA) clubs, as well as to foster communication and provide information in a manner that is both simple and efficient. We will be working with SPA to gain an understanding of the club workflow process so we can best integrate our software with existing systems. We will orchestrate a system architecture to include mobile applications that can be deployed on android devices. These devices will communicate with our proxy server via API Requests. These requests will be processed by the proxy server and sent to the Database via MySQL Scripts. The application will have multiple different user levels: non-member, member, leader, and admin. User levels are not universal but are club dependent (a user can be a member in one club, a president in another club, and a non-member in any other club). The application will be used for searching for clubs, finding out information for clubs, leaders posting announcements/polls/events in a club page, inter-club communication, admins (SPA users) can create, approve, and delete clubs. Additional Features will be added as needed.

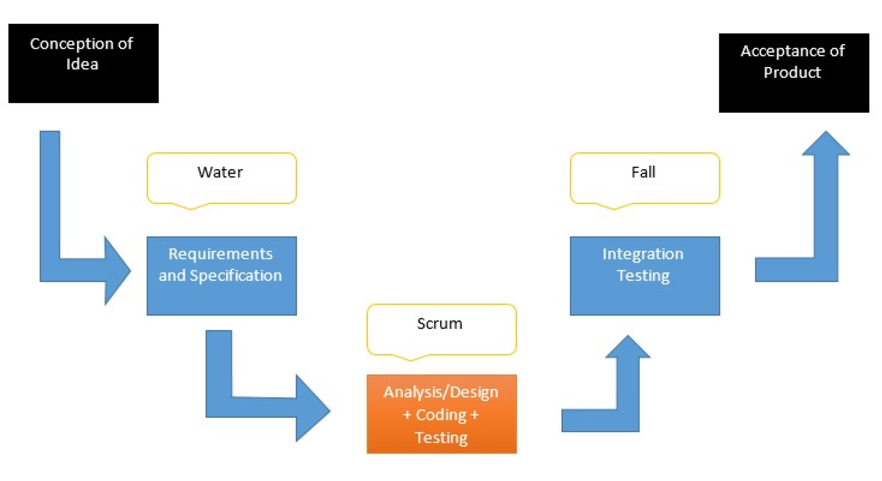
### Project Deliverables

* Android Application UI
  + Prototype UI (developed in Android Studio)
  + Final UI (sideloaded on Android devices)
* MySQL database
  + Initial MySQL database (tested locally)
  + Final MySQL database (implemented on Google Cloud instance)
* Android Application Functions
  + Prototype application with some features (incomplete implementation)
  + Final application with all features (complete implementation)
* Proxy Server Infrastructure
  + Prototype Proxy Server Functionality (simple communication between Application, Proxy Server and Database)
  + Final Working System (complete integration of Application, Proxy Server, and Database)

# Project Organization

### Software Process Model

* **Water-Scrum-Fall Model**



### Roles and Responsibilities

* **Scrum Product Owner**
  + - Stephen Styffe
* **Scrum Master / Project Manager**
  + - Hannah Kim
* **Configuration Manager**
  + - Zachary Chester
* **Developers**
  + - Jiahui Jiang
    - Wesley Shiozaki
    - Zachary Chester
* **Testers**
  + - Jiahui Jiang
    - Wesley Shiozaki
    - Zachary Chester

### Tools and Techniques

* **Azure DevOps**
  + - Project Manager Tool to Manage Scrum Backlogs and Task Assignments
* **Google Cloud**
  + - Backend Cloud Environment to Run Proxy Server in
* **Github**
  + - Project Code Repository to Maintain Version Reliability in Development
* **Gmail SMTP server**
  + - Email Functionality through API Calls from the Proxy Server
* **MySQL**
  + - Database Backend Console
* **Android Studio**
  + - Front-end Mobile Application Development Environment
* **Navicat Data Modeler Essentials**
  + - Database Modeling for Database Design

# Project Management Plan

## Tasks:

### Task 1 - Setting up Android Studio Project

Description

* Create blank Android Studio project
* Connect project with Github
* Connect project with database

Deliverables and Milestones

* Android Studio project that is connected with Github and database

Resources Needed

* Github account
* A setup database

Dependencies and Constraints

* High performance Windows machines

Risks and Contingencies

* Potential for errors in creating/connecting

### Task 2 - Setting up Azure Table

Description

* Design and implement a server to act as a proxy between various Android devices and MySQL database using APIs and MySQL scripts for secure database, being accessed by the one server rather than every Android application

Deliverables and Milestones

* Functional Linux instance that houses MySQL database, python API scripts, and MySQL scripts

Resources Needed

* Google Cloud Console with Compute Engine instance
* Google Cloud SDK

Dependencies and Constraints

* Project Budget
* Application Backend Design
* Database Model

Risks and Contingencies

* Limited budget
* Potential for errors regarding database and/or backend design
* Potential for access errors

### Task 3 - Create Proxy Server on gCloud instance

Description

* Design a Server Architecture Model
* Implement a Functional Proxy Server between the Database and Application

Deliverables and Milestones

* Linux Instance that Houses MySQL Database
* Python API Scripts
* MySQL Scripts

Resources Needed

* Google Cloud Console with Compute Engine instance
* Google Cloud SDK Shell

Dependencies and Constraints

* Project Budget - Infrastructure Requirements
* Application Backend Design
* Database Model

Risks and Contingencies

* Limited Budget
* Errors in the Database
* Errors in the Backend Design
* Network Access / Server Request Errors

### Task 4 - Design MySQL DB

Description

* Create ER diagram to model relation
* Design and create MySQL database for needs of application

Deliverables and Milestones

* Entity Relationship diagram
* MySQL database

Resources Needed

* Navicat Data Modeler Essentials
* MySQL

Dependencies and Constraints

* Google Cloud instance

Risks and Contingencies

* Potential for errors in database
* Potential for unintentional modification/deletion of data in database

# Assignments

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | R = Responsible | A = Accountable | C = Consulted | I = Informed | Q = Quality Reviewer |
|  | **John** | **Stephen** | **Hannah** | **Wesley** | **Zach** |
| **Android Studio (Java)[[1]](#footnote-0)** | R | I | C |  | A |
| - UX Navigation & Workflows | R | I | A | Q | Q |
| - UI Design & Implementation | R | I | C | Q | Q |
| - Front-Backend Binding | R | I | Q | Q | C |
| - Backend (Classes) | C | I | R | C | C |
| **Proxy Server (Python)** |  | C | Q | Q | R |
| - API scripts | C | C | C | Q | R |
| - Database query scripts |  | R | Q | C | C |
| - Server setup and config |  | C | Q | Q | R |
| **Database (MySQL)** |  | R | A | Q | I |
| - DB Modeling |  | R | C | C | C |
| - DB Implementation |  | R | Q | C | I |
| **Planning / Schedule** | A | A | R | Q | I |
| **Risk Management** | A | A | Q | R | I |
| **Quality Management** | A | R | A | Q | I |

# Timetable[[2]](#footnote-1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Name** | **Responsible** | **Start** | **End** | **Man Hours** | **Status** |
| SPMP | all | 1/22 | 1/25 | 20 | In Progress |
| SRS | all | 1/26 | 2/1 | 15 | Not started |
| Scrum Backlog | Hannah | 2/5 | 2/8 | 6 | Not started |
| SDD (Design) | all | 2/2 | 2/15 | 32 | Not started |
| DB Design | Stephen | 2/2 | 2/15 | 7 | Not started |
| System Architecture Design | Zach | 2/2 | 2/15 | 10 | Not started |
| Application Design | John | 2/2 | 2/15 | 15 | Not started |
| Sprint1 | all | 2/15 | 3/1 | 65 | Not started |
| MySQL DB Creation | Stephen | 2/15 | 3/1 | 20 | Not started |
| Instance Setup | Zach | 2/15 | 3/1 | 15 | Not started |
| Android Studio Back End | John | 2/15 | 3/1 | 30 | Not started |
| Sprint2 | all | 3/2 | 3/25 | 75 | Not started |
| MySQL Scripts Creation | Stephen | 3/2 | 3/25 | 20 | Not started |
| API Scripts Creation | Zach | 3/2 | 3/25 | 25 | Not started |
| Android Studio Front End | John | 3/2 | 3/25 | 30 | Not started |
| Sprint3 | all | 3/26 | 4/5 | 48 | Not started |
| DB Review / Maintenance Plan | Stephen | 3/26 | 4/5 | 8 | Not started |
| SMTP Implementation | Zach | 3/26 | 4/5 | 10 | Not started |
| Android Studio Additional Features | John | 3/26 | 4/5 | 20 | Not started |
| System Review | Zach | 3/26 | 4/5 | 10 | Not started |
| SUSTD | all | 4/6 | 4/22 | 30 | Not started |

# Additional Material

### Additional Issues

None so far

### Definitions, Acronyms, and Abbreviations

* Scrum - a set of practices used in agile project management that emphasize daily communication and the flexible reassessment of plans that are carried out in short, iterative phases of work.
* Sprint - a regular, repeatable work cycle in scrum methodology during which work is completed and made ready for review
* Student Programming & Activities (SPA)
* Database (DB)
* Entity Relationship (ER)
* Software Project Management Plan (SPMP)
* Software Requirements Specifications (SRS)
* Software Design Description (SDD)
* Software Unit and System Test Documentation (SUSTD)
* Simple Mail Transfer Protocol (SMTP)
* Application Program Interface (API)

### References

None so far

### Appendices

None so far

1. Bold assignments show assignment for the entirety of the task, while the presence of non-bold subtasks allow delegation of large tasks. [↑](#footnote-ref-0)
2. note: each colored row is a summation of the following subtasks. [↑](#footnote-ref-1)