Software Requirements Specification  
Scrum Manager

|  |  |
| --- | --- |
| Ryan Feeny | ryanq.feeney@ufl.edu |
| Jesse Martinez | kroronos@ufl.edu |
| Tony Strother | semphy@ufl.edu |
| Stephen Berkner | sberkner@ufl.edu |
| Kyle Collins | Kcoll196@ufl.edu |
| Bryan Fallin | fallinbryan@ufl.edu |

Table of Contents

[**1 INTRODUCTION** 3](#_Toc504338418)

[**1.1 Purpose** 3](#_Toc504338419)

[**1.2 Project Summary** 3](#_Toc504338420)

[**1.3 Target Platform(s) / Operating System(s) Supported** 3](#_Toc504338421)

[**1.4 Tools/APIs/Development Environment/Programming Languages** 3](#_Toc504338422)

[**2 SYSTEM DESCRIPTION** 3](#_Toc504338423)

[**2.1.1 (Example) Application Modes** 3](#_Toc504338424)

[**2.1.2 (Example) Database description** 3](#_Toc504338425)

[**2.1.3 (Example) Network description** 3](#_Toc504338426)

[**2.2 User Characteristics** 3](#_Toc504338427)

[**2.3 Constraints** 4](#_Toc504338428)

[**2.4 Wireframes** 4](#_Toc504338429)

[**3 FUNCTIONAL REQUIREMENTS** 4](#_Toc504338430)

[**3.1 Main Features** 4](#_Toc504338431)

[**3.2 Use Cases** 4](#_Toc504338432)

[**3.3 Use case diagrams** 4](#_Toc504338433)

[**4 EXTERNAL INTERFACE REQUIREMENTS** 4](#_Toc504338434)

[**4.1 User Interfaces** 4](#_Toc504338435)

[**4.2 Hardware Interfaces** 4](#_Toc504338436)

[**4.3 Software Interfaces** 4](#_Toc504338437)

[**4.4 Communications Interfaces** 5](#_Toc504338438)

# **1 INTRODUCTION**

### **1.1 Purpose**

SRS version 1.0 for Group16 Scrum Manager

### **1.2 Project Summary**

A project management interface based on the scrum development process.

### **1.3 Target Platform(s) / Operating System(s) Supported**

* Windows
* Linux
* Mac

### **1.4 Tools/APIs/Development Environment/Programming Languages**

* This project will by using Python 3.6 32bit, the included tkinter library, MySQLdb library, and the PyGitHub Library
  + <https://www.python.org/downloads/release/python-360/>
  + <https://stackoverflow.com/questions/372885/how-do-i-connect-to-a-mysql-database-in-python>
  + https://github.com/PyGithub/PyGithub
* IDE will be Jetbrains PyCharm
  + https://www.jetbrains.com/pycharm/
* Version Control will be managed through GitHub
  + <https://github.com/CEN3031-group16/GroupProject>

# **2 SYSTEM DESCRIPTION**

This system will allow teams to visually manage development projects, track development progress, and communicate efficiently through an intuitive GUI

System will connect to a remote MySQL server @ 173.230.136.241 to store project data so that multiple users across different workstation and platforms can work on the same project with synchronized data. User PC must be connected to the internet for the application to function.

System will model the Scrum agile development process. Backlog items will be stored in the database as cards. that will have foreign keys associated with an assigned user, a sprint cycle, and a link to explore code hosted on GitHub. Cards will have type ‘Epic’, ‘User Story’, ‘Feature’, ‘Bug’, ‘Re-Factor’, and ‘Complete’. Cards will be represented in code as distinct objects in a map hashed by a card’s unique table key.

Users will have privilege levels based on role, only admins can create new projects, new users, assign user roles and everything below. Scrum masters can finalize backlog items as complete, move cards back to the backlog for bugs or refactoring, initiate Sprints, assign cards to users, lock a card to a dev user, and everything below. Dev users can assign cards to themselves, comment on cards, create new cards, submit cards for review, link cards to code, un-assign themselves from cards unless locked, and change their own password.

Sprint Cycles will be stored in the database with a unique key, start date, and due date. Assigning cards to sprint cycles will occur by populating a card sprint cycle field with the unique key. If a user is assigned to a card that is assigned to a sprint, that user is assigned to the sprint.

Comment will be stored in the database in a comment table with a unique identification key, a timetamp, and foreign keys that associated with users, cards, and sprints.

Users will be required to authenticate upon opening the application. Login credentials will not be cached locally.

Start up GUI will be a dashboard that displays current sprint details in the center, backlog on the left, list of all comments ordered by date on the right. Center display will be different based on user role.

### 

### **2.1.1 Application Modes**

Application will have user specific Mode

|  |  |
| --- | --- |
| Admin Mode | Can Create and delete projects and users |
| Scrum Master Mode | Can Assign User Roles |
| Developer Mode | Limited Admin Privileges |

### **2.1.2 Database description**

Remote MySQL server

User Project\_Name Database

User Table

Int: User ID

Text: User Name

Text: User email address

Text: User password (encrypted)

Text: User Role

Text: Github user ID

Text: GitHub user password (encrypted)

Card Table (These are backlog items)

Int: Card ID

String: Card Type

Int: Priority

Foreign Key: Assigned User ID

Text: Card Title

Text: Card Description

Date: Due Date

Date: Assigned Date

Foreign Key: Sprint ID

?? : GitHubRepo->Project->Class

Int: Status ( 0 = unassigned; 1 = in progress; 2 = completed)

Sprint Table:

Int: Sprint ID

Date: Start Date

Date: Due Date

Comment Table:

Int: Comment ID

Date: Comment Timestamp

Foreign Key: Card ID

Foreign Key: User ID

Foreign Key: Sprint ID

### **2.2 User Characteristics**

The typical user is a member of a development team utilizing the Scrum process for product development.

### **2.3 Constraints**

Application requires a pre-configured remote MySQL database

Linking code to a card will requires a user GitHub account

### **2.4 Wireframes**

Provide wireframes of major features to illustrate the overall structure/flow of the application. (At least 1 wireframe per person in the group).

# **3 FUNCTIONAL REQUIREMENTS**

### **3.1 Main Features**

* GUI
* Visual Backlog Queue
* Create Users
* Create Backlog Items
* Create Sprints with start dates and due dates
* Drag users to Backlog Items to assign tasks
* Drag Backlog Items to Sprints
* Auto create Sprints bases on Backlog priority Queue
* Add comments to Backlog Items or Sprints
* View Latest Code associated with Backlog Item From GitHub
* Connect to database
* Connect to Git repo

### **3.2 Use Cases**

Create a description of use cases for several of the features in your project.

### **3.3 Use case diagrams**

Create use case diagrams to illustrate several use cases.

# **4 EXTERNAL INTERFACE REQUIREMENTS**

### **4.1 User Interfaces**

*Specify the logical characteristics of each interface between the software product and its users (e.g., required screen formats, report layouts, menu structures, or function keys).*

*Specify all the aspects of optimizing the interface with the person who must use the system (e.g., required functionality to provide long or short error messages). This could be a list of do’s and don’ts describing how the system will appear to the user.*

### **4.2 Hardware Interfaces**

*Specify the logical characteristics of each interface between the software product and the hardware components of the system. This includes configuration characteristics (e.g., number of ports, instruction sets), what devices are to be supported, and protocols.*

### **4.3 Software Interfaces**

*Specify the use of other required software products (e.g., a database or operating system), and interfaces with other application systems.*

*For each required software product, provide identification information including at least name, version number, and source.*