**4 Bored & Broke Software Development Plan**

**4.1 Plan Introduction**

This Software Development Plan provides the details of the planned development for the Bored & Broke web application which allows users to connect with other users based on their similar interest in a certain activity.

On the Bored & Broke website, the user will be asked to create a profile, which will store information such as location, name, and saved activities. The user can then see and save activities in their location. When a user saves an activity, they are brought to that activity’s page where they are then able to contact those users individually. This application is similar to Tinder, Bumble, and Meetup. It allows users to connect with people they may not know based on mutual interests and location. It is different because it “matches” people together based on their interest in activities that already exist (Meetup requires users to create their own events/activities). This caters to our end user, who is a person who is bored, broke, and alone (and doesn’t want to be).

By 10/18, the homepage will be able to display activities that have been requested from various API’s. By 10/25, the database schema will be defined and the login/create account web page will be created. By 11/1 the Software Development Document will be written. By 11/8 users will be able to log in and out of accounts that store their information. The alpha beta presentation will then be on 11/29.

**4.1.1 Project Deliverables**

**9/20/17:** *Project Proposal Document* - This document is made up of two parts. First is a description of what the project is, which is high-level overview of the project detailing software/hardware requirements along with the future vision of the project. The second part is the project justification. This section serves to explain why the project is appropriate for the class in terms of use of knowledge from previous classes, suitable amount of difficulty, etc.

**9/27/17:** *Requirements Specification* - This document outlines the low level requirements for the project: functional, performance, and environment requirements. Initial Development Schedule (Part of SDP draft) - This is a preliminary project schedule that gives the dates for project deliverables.

**10/11/17:** *Software Development Plan Document (complete)* - This documents gives a detailed description of the entire project. This includes explaining the process that will be used to produce the required software and final product by the end of the semester. It will also discuss specific member roles, team structure, as well as the schedules for team meetings and project deliverables.

**11/1/17:** *Software Design Description Document (Architecture Section)* - This part of the document consists of the architectural, or in other words, overall design of the project. This is done by breaking the design into classes, which are based off the functionalities of the application.

**11/15/17:** *Software Design Description Document (Complete)* - This will include the second part of the document that includes a detailed design of the application through use of text and diagrams.

**11/29/17:** *Test and Integration Plan* - This plan will introduce the software testing strategies, outline the unit testing requirements, and describe the unit and integration tests for the application.

*ALPHA/BETA Presentation/Demonstration* - Functionality of the application will be extremely close, if not completed. Minor design adjustments may need to be made; however, the application should be usable. A draft of the User’s Manual will also be provided. This manual will explain how to use every feature and function of the application.

**12/13/17:** *User's Manual Final Updates* - Completed version of the User’s Manual with any updates made since the Alpha/Beta demonstration.

*Final Product Delivery* - Application will be provided with completed functionality and design.

**4.2 Project Resources**

**4.2.1 Hardware Resources**

| **Resource** | **Development** | **Execution** |
| --- | --- | --- |
| macOS Machine | ✓ | ✓ |
| Windows Machine | ✓ | ✓ |
| Linux Machine | ✓ | ✓ |
| 2 GB RAM (Minimum) | ✓ | ✓ |
| Display | ✓ | ✓ |
| Internet Connection | ✓ | ✓ |
| Intel Processor |  | ✓ |
| ARM Processor |  | ✓ |

**4.2.2 Software Resources**

| **Resource** | **Development** | **Execution** |
| --- | --- | --- |
| macOS | ✓ | ✓ |
| Windows 10 | ✓ | ✓ |
| Text Editors (Atom, Brackets) | ✓ |  |
| Web Broswers (Chrome, Firefox, IE) | ✓ | ✓ |
| Node.js | ✓ | ✓ |
| React.js | ✓ | ✓ |
| Postgres | ✓ | ✓ |
| Heroku | ✓ | ✓ |
| Bootstrap | ✓ | ✓ |
| GitHub | ✓ |  |
| Adobe Experience Design (XD) | ✓ |  |
| Slack | ✓ |  |
| Google Drive | ✓ |  |

**4.3 Project Organization**

The following table shows members for our three different teams: Backend, Frontend, and Database. While we have assigned teams, we expect to be moving throughout the teams frequently.

| **Team Member** | **Teams** |
| --- | --- |
| Mary | Backend, Database |
| Jordan | Frontend, Backend |
| George | Backend, Database |
| Mat | Backend, Database |
| Josh | Frontend |
| Mackenzie | Frontend |
| Megan | Frontend |

**Team Responsibilities**

**Frontend:** wireframe designs, mockup designs, frontend designs, Heroku/Deployment, HTML

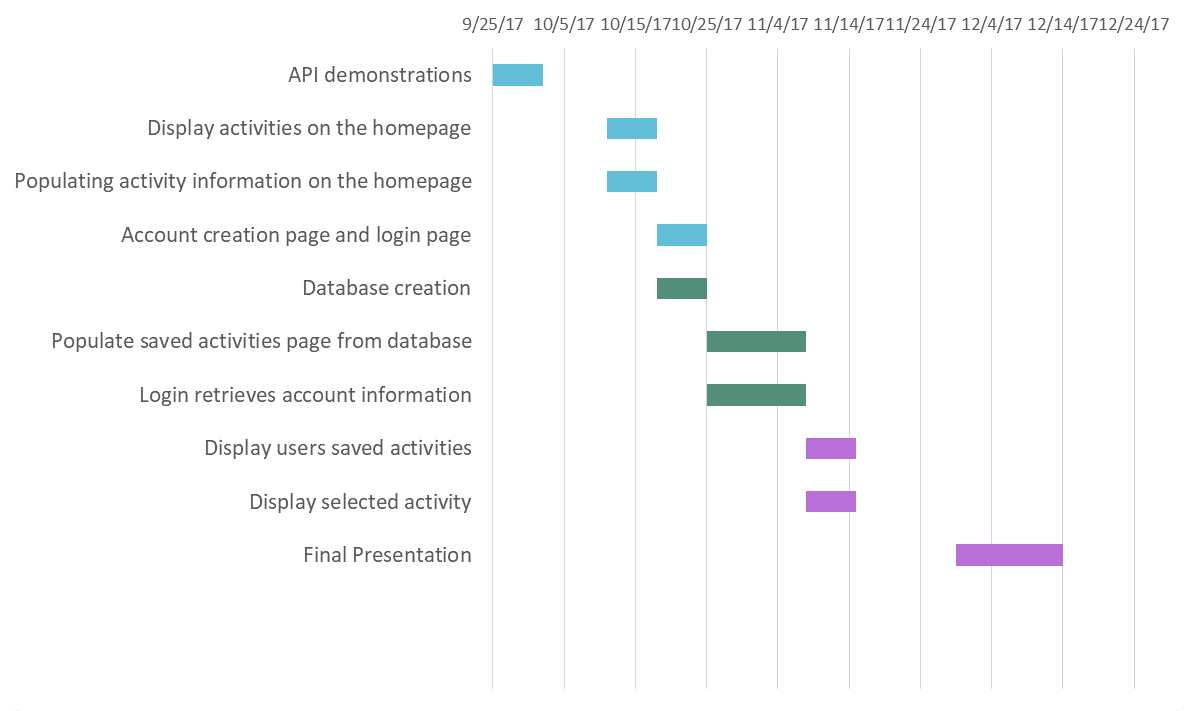
**Backend:** JSON object definitions, database integration, API calls, account verification

**Database:** schema definition, query writing

**4.4 Project Schedule**

Our project schedule consists of our class deliverables as well as software deliverables toward the end product.

**4.4.1 GANTT Chart**

[](NULL)

**4.4.2 Task/Resource Table**

| **Task** | **Team** | **Hardware** | **Software** |
| --- | --- | --- | --- |
| Mockups | Frontend | Macbook | Adobe Experience Design |
| API demonstrations | All | Macbook, HP Pavilion | Atom, Github, Chrome, Safari |
| Display activities on the homepage | Frontend | Macbook | Atom, Github, Chrome, Safari, Heroku, Node.js |
| Populating activity information on the homepage | Backend | Macbook, Hp Pavilion | Atom, Github, Chrome, Safari |
| Account creation page and login page | Frontend | Macbook | Atom, Github, Chrome, Safari, Heroku, Node.js |
| Database creation | Database | Macbook, HP Pavilion | Atom, Github, Chrome, Safari, Node.js, PostgreSQL, Sequelize.js |
| Populate saved activities page from database | Backend | Macbook, HP Pavilion | Atom, Github, Chrome, Safari, Node.js, PostgreSQL, Sequelize.js |
| Login retrieves account information | Backend | Macbook, HP Pavilion | Atom, Github, Chrome, Safari, Node.js, PostgreSQL, Sequelize.js |
| Display users saved activities | Frontend | Macbook | Atom, Github, Chrome, Safari, Heroku, Node.js |
| Display selected activity | Frontend | Macbook | Atom, Github, Chrome, Safari, Heroku, Node.js |

Nice work on this!

5 out of 5% = A-plus

Nothing else needed for this.