4.1 SDP Plan Introduction

This software development plan provides the details of the planned development for a fitness tracking web application. It will allow users to create an account where they can log their workouts and goals. It will track their progress over time and display their milestones. The main justification for this project is the lack of any existing software as a web application. Software with similar features exist on mobile devices, but not as a web application. Any web applications that are similar either cost a subscription and/or complicate the process. The goal of this project is to simplify the features to what athletes actually want to use, using an efficient backend and a well-designed user interface that is both intuitive and aesthetically pleasing.

The development process can be broken down into two main phases. The first is building the backend. This will allow users to create accounts and store and manipulate their data. This will include creating an API using Nodejs and a Database using Neo4j. The second phase is creating the frontend. This will provide the user with a way to access all of this information in the form of a web application. After these two phases are completed, the remaining time for development will be spent styling the application and testing it.

|  |  |  |
| --- | --- | --- |
| Item | Description | Date |
| Backend | API and Database | Week 13 |
| Frontend | Front end web app | Week 14 |
| Testing and Styling | Testing the app as a whole. Improving on the UI | Week 16 |
| Release | Project finished and released | Week 17 |

4.1.1 Project Deliverables

* Backend
  + Completed by Week 13
  + Includes Nodejs API
  + Includes Neo4j Database
* Frontend
  + Completed by Week 14
  + Will use HTML, CSS, and JavaScript
  + Will use Bootstrap for styling
  + Will be able to create and edit content stored in our backend
* Testing and Styling
  + Completed by Week 16
  + Testing
    - Testing CSS including resizing of windows and displaying data
  + Styling
    - Focus on Bootstrap and CSS improvements
    - Will have final emblem and application name
* Release
  + Completed by Week 17
  + Will include all final changes to Backend and Frontend
  + Final product

4.2 Project Resources

Resources are elements that will be managed, but will assist in the completion of the project. The following subsections will overview the different resources, which will be used during development.

4.2.1 Hardware Resources

4.2.1.1 API and Database

4.2.1.2 Web Application

The hardware resources needed for building the web application are the same that are required for the user to run it.

|  |  |
| --- | --- |
| Category | Requirement |
| Operating System  Browser  Processor  Graphics | Mac OS X  Google Chrome  Safari  Intel Core i7  Intel HD Graphics 4000 1024 MB |

4.2.2 Software Resources

4.2.2.1 API

* Node.js v0.12.0
* Neo4j v2.2

4.2.2.2 Web Application

* HTML
* CSS
* JavaScript
* Bootstrap v3.3.2

4.2.2.3 Entire Project Creation

* Sublime Text
* GitHub for Mac

4.3 Project Organization

This project can be clearly separated into two distinct “teams”. The first being API and the second being Web Application. Both of these teams or components make up the full stack of this system. Each have distinct roles that are outlined below.

4.3.1 API

Since this application is based around storing and retrieving data, the API is essential for any sort of functionality. This is why it is taking precedence over the Front End. The tasks to be handled, in their respective order, are listed below.

1. Account creation
2. Workout
   1. Log
   2. Edit
   3. Delete
3. Goal
   1. Log
   2. Edit
   3. Delete
4. List goals and workouts for viewing/editing/deleting

4.3.2 Web Application

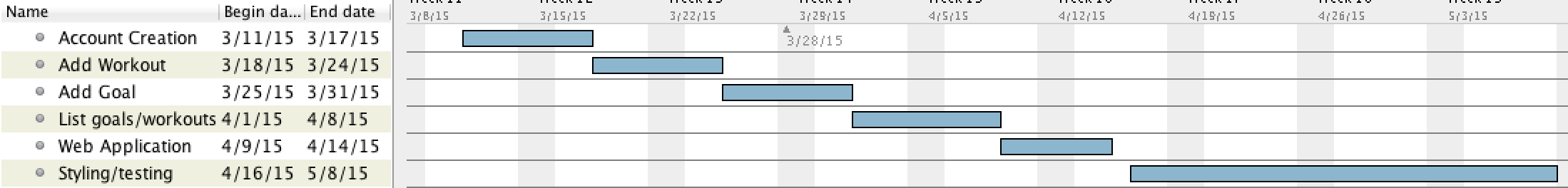
The web application is what the user will be interacting with. It will be what displays the data for the user and allows them to interact with and manipulate it. The tasks, in order, are listed below.

1. Signup/Login View
   1. Be able to enter information and create account
2. Log entry view
   1. Be able to enter and save a workout
3. Goal entry view
   1. Be able to enter and save a goal
4. Log list view
   1. Display past workouts that have been saved by user
   2. Be able to edit these workouts and save the changes
   3. Be able to delete a workout
5. Goal list view
   1. Display goals the user has set
   2. Be able to edit, delete, or check goal completed.
6. Profile view
   1. Display user information
7. Settings view
   1. Display current settings
   2. Have buttons that allow user to edit settings

4.4 Schedule

This section provides schedule information for the project

4.4.1 Gantt Chart



4.4.2 Task/Resource Table

|  |  |
| --- | --- |
| Task | Resource |
| API - Account Creation | Neo4j, Node.js, Sublime Text, Github |
| API - Add Workout | Neo4j, Node.js, Sublime Text, Github |
| API - Add Goal | Neo4j, Node.js, Sublime Text, Github |
| API - List Goals/Workouts | Neo4j, Node.js, Sublime Text, Github |
| Web Application | Bootstrap, Sublime Text, Github |
| Styling/Testing | Bootstrap, Sublime Text, Github |