Project: Calendar Scheduler application

By B A B I L S H I A . J

SREELEKHA.L

RITHIKA.S

MAR 05, 2023

Contents

1 Introduction

This report consists of three software development documents. These documents include: Vision and Scope, Software Requirements Specification, and Software Architecture Design. The application that will be discussed in the following sections is not only meant to be a standalone application, but also an application that can be plugged into a larger project management application. With this being said, the portion of the project management application that is the focus of the following documents and the final product application is a calendar task tracking application.

The creation of each document is intended to further better the vision of the final product. In the software industry, the Vision and Scope document is meant to ensure that the stakeholders and engineers share a common understanding of the goal they want to reach. The Software Requirements Specification (SRS) defines what features an application will provide and some of the purposes for its creation. Finally, the Architecture Design Documentdescribes how separate components of the system work together.

In general, this calendar application is meant to help individuals with busy schedules. Those who must balance their time between working on multiple simultaneous projects would categorize key users. In the application, users will be able to participate in multiple projects at any one time. When a user is in a project, they will be able to create tasks, add tasks, be assigned to tasks, and add other users to the project. A key feature in this application is that each user is provided a personal project. In their personal project, any task assigned to the user from any project, will be synced to this project calendar. This will allow the user to easily view the tasks they need to complete given the start and end dates.

2 Vision and Scope

2.1 Business Requirements

2.1.1 Background

Projects are present in every industry. As projects can be broken into several tasks, it can be difficult to keep track of them all without some tools to help keep track of them for us. A planner is one of these tools; they help remind an individual of both their personal and project tasks, however physical planners are limited. They only provide a certain amount of space to detail the tasks for any given day. An application that provides a digital calendar/planner eliminates the limitation of physical space in a physical planner.

2.1.2 Business Opportunity

There is an opportunity for a project management application that will allow users to track their project tasks. For projects completed by multiple individuals, this application will be a tool to share all of the project's tasks between the users. The application will also eliminate the need for a separate personal planner by providing one for the user that syncs their project tasks to be displayed alongside their personal tasks.

2.1.3 Business Objectives and Success Criteria

BO-1	To build an application that allows all participants to keep track of a project's tasks
BO-2	To build an application that tracks all of the user's tasks in one location
BO-3	To provide tools necessary for project success in one application
SC-1	Customers will want to use this application over other project management and calendar applications for the convenience it offers

2.1.4 Customer or Market Needs

CN-1	An application comparable in usability to popular project management or calendar applications with the convenience of all the necessary tools for project success in one location

2.1.5 Business Risks

The primary risk is that the market for project management applications is already full of solutions. Therefore, creating a viably successful and competitive solution will be difficult.

2.2 User Description

2.2.1 User/Market Demographics

Individuals with a busy schedule make up the primary demographic for this application. To specify, individuals who take part in one or more projects, or even an overwhelming number of non-project related tasks make up the demographic aim for this application.

2.2.2 User Personas

The primary personas include:

1. A busy industry professional working on a large project.

2. A college student with a large workload, that includes school and a part time job.

The secondary persona consists of an individual who wants to keep track of all their tasks through a digital planner.

2.2.3 User Environment

The application will primarily be accessible on the web. Therefore, users should be able to access the application on any device connected to the internet.

2.2.4 Key User Needs

1. Ability to create and view a project represented by a calendar.

2. Ability to add users to view a project.

3. Ability to add tasks to a calendar.

4. Ability to add task details.

5. Ability to view a personal calendar.

6. Ability to sync project tasks with the personal calendar.

2.3   Vision of the Solution

2.3.1   Vision Statement

The intent of the application is to make project management more convenient and streamlined.

2.3.2   Solution Overview

The application will begin focused as a digital calendar. Once a project is created, users can view and add tasks related to the project. All users with access to a project will be able to perform the aforementioned actions. Users will also have access to their private personal calendar, where they can add personal tasks that are unrelated to their other projects.

2.4   Business Context

2.4.1   Stakeholder Profiles

Name	Description and Background
Project Members	Project members are individuals participating in a project that is broken up intoseveral tasks.

2.4.2   Project Priorities

1. Ability to create and view a project represented by a calendar.

2. Ability to add users to view and edit a project.

3. Ability to add tasks to a calendar.

2.4.3   Operating Environment

The application will be aimed as a web application and should be able to run on any device connected to the internet.

2.5   Competitive Analysis

2.5.1   Overview

There are several different competitors for project management and calendar applications that already offer project management tools in one convenient application, and/or a calendar which keeps track of tasks. This section will cover two competitors, the first is a project management tool, while the second is strictly a calendar application.

2.5.2   Competitor 1 - Asana

Asana offers a wide variety of project management tools to keep teams organized in one easy to use platform. Asana allows users to create a task and provide task details. It also offers a team calendar as well as a conversation feature. Users can view tasks grouped by those assigned to them or those they have assigned to others. Asana provides great tools for project management; however, it still lacks helpful features such as documentation features. Although the proposed application will offer similar features to Asana, its advantage comes in its design which will allow for easy addition of futureproject management tools.

2.5.3   Competitor 2 - Google Calendar

Google Calendar offers its users a way to keep track of their tasks. It is a great tool for those with a busy schedule; users can set their calendar to display tasks by the hour. Users also have the ability to set appointments and can share their calendar with others. They also have the ability to make multiple calendars if they so desire. Although Google Calendar is a flexible and useful tool, it is not meant specifically for team projects. Google Calendar also lacks other project management tools. Google itself offers other solutions for project management and documentation, however these tools are not all accessible from one application. Each tool is a standalone application.

3 Software Requirements Specification

3.1 Introduction

This SRS serves as a description of the software application to be built for release 1.0 of the Calendar Application. The following provides functional requirements in the form of use cases, system features, external interfaces, and nonfunctional requirements. This document will be used to verify the correctness of the application upon completion of version 1.0.

3.1.1 Document Conventions

The following standards are used in this document:

1. Knuth's Computer Modern 12pt font.

2. All writing is written in third person.

3.2 Intended Audience and Reading Suggestions

3.2.1 Developer

Developers will oversee analysis, design, implementation, integration, and testing of all artifacts produced for the application. Developers will reference this document to make sure that they are complying to all functional and nonfunctional requirements outlined in this SRS and to make sure that the vision of the application is being fulfilled.

3.2.2 Supervisor

We will use this document to obtain the intended vision for this application. Suggested Readings:

1. Section 2: Overall Description

2. Section 3: Use Cases

3.3 Project Scope

The purpose of this project is to place all tasks for an individual's multiple projects in one location. Additionally, the application should allow users to easily collaborate on projects by allowing each member of the project to view all the tasks for the project, no matter who it's assigned to. For more information, please refer to the Vision and Scope document.

3.4 References

The following document(s) should be used as an aid to this SRS document:

1. Vision and Scope Document

3.5 Overall Description

3.5.1 Product Perspective

View the" Vision of the Solution" section located in the Vision & Scope document.

3.6 Product Significant Functions

1. Ability to create a project

2. Ability to add tasks to a project

3. Ability to add users to a project

4. Ability to view all tasks pertaining to a project the user is a part of

3.6.1 Operating Environment

The application is planned to run on any device with an internet connection. The application's design should be focused around being extensible.

3.6.2 Design and Implementation Constraints

The application and its developers have flexibility about specific features and implementation. The application does not have a set limit of total users or traffic and must be able to scale accordingly. Security concerns are

centralized around user accounts which will be managed and authenticated through the use of a third-party system. This project and its development must be managed with an Agile approach.

3.6.3   User Documentation

User documentation will be minimal as the application will be structured much like other calendar applications. Users already familiar with calendar and project management applications should be able to readily understand how this application functions.

3.6.4   Assumptions and Dependencies

1. Use of a third-party user authentication service will improve the service and reduce coding time and effort.

2. Maintaining aspect ratio across multiple devices will not be a significant challenge.

3. Users will be able to use the application on both mobile and desktop devices equally well.

4. Projects will not require an official administrator or moderator.

3.7 Use Cases

All the following use cases assume the user has signed into the application as a precondition.

3.7.1 Use Case 1: Create a New Project

Use Case ID:	1
Use Case Name:	Create a New Project
Created By:	Babilshia, Sreelekha, Rithika
Last Updated By:	Babilshia, Sreelekha, Rithika
Date Created:	March 05,2023
Date Last Updated:	March 10,2023
Actors:	User
Description:	User can create a new project and view the project calendar
Preconditions:	
Postconditions:	1. User can see the project calendar focused on the current month.
Normal Flow:	1.0 Create a new project 1. User clicks the new project button. 2. A pop-up window for a new project appears. 3. The user enters a name for the project. 4. The user clicks the create button. 5. The pop-up window disappears, and a new project calendar becomes visible in the calendar portion of the screen. The project calendar displays the current month. Among the list of projects is a new tab with the provided project name.
Alternative Flows:	1.1 User cancels operation 1. Branch after step 2 of normal flow. 2. User clicks the cancel button. 3. The pop-up window disappears, and the screen remains unchanged from when the user clicked the new project button.

	1.2 User adds other users to the project
	1. Branch after step 3 of normal flow. 2. The user enters the username of other users. 3. Return to step 4 of normal flow.
Exceptions:	None
Priority:	High
Frequency of Use:	Very Common

3.7.2   Use Case 2: Add a New Task

Use Case ID:	2
Use Case Name:	Add a new task
Created By:	Babilshia, Sreelekha, Rithika
Last Updated By:	Babilshia, Sreelekha, Rithika
Actors:	User
Description:	User can add a new task to either a project or their personal calendar
Preconditions:	1. User has either their personal calendar or a project calendar open.
Postconditions:	1. User can see the start date, specified for the task, on the calendar is shaded.
Normal Flow:	1.0 User adds a new task to the calendar 1. The user clicks the new task button. 2. A pop-up window for the new task appears. 3. The user enters a task title, an assigned user. 4. The user must enter a start date, an end date, and a description. 5. The user clicks the create button. 6. The pop-up window disappears, and the screen returns to the screen prior to the user clicking the new task button.
Alternative Flows:	1.1 User cancels the new task

	1. Branch after step 2 of normal flow.
	2. The user clicks the cancel button.
	3. Return to step 6 of normal flow.
	1.2 User does not provide a task title or assigned user.
	1. Branch after step 2 of normal flow.
	2. The user clicks the create button.
	3. A warning displays that the user must provide a task title and assigned user.
	4. Return to step 3 of normal flow.
Exceptions:	None
Priority:	High
Frequency of Use:	Very Common

3.7.3 Use Case 3: Switch between project and personal calendars

Use Case ID:	3
Use Case Name:	Switch between project and personal calendars
Created By:	Babilshia, Sreelekha, Rithika
Last Updated By:	Babilshia, Sreelekha, Rithika
Actors:	User
Description:	The user switches from a project calendar to their personal calendar. All project tasks assigned to the userappear in their personal calendar.
Preconditions:	1. User has an existing project calendar.
Postconditions:	1. User is viewing their personal calendar filled with their project tasks.
Normal Flow:	1.0 User views a synced project task on their personal calendar

	1. The user has a project calendar open.
	2. The user creates a new task for the project assigned to themselves.
	3. The user switches to their personal calendar by clicking their personal project button.
	4. The user's screen changes to their personal calendar and the newly created task is visible.
Alternative Flows:	None
Exceptions:	None
Priority:	High
Frequency of Use:	Very Common

3.7.4   Use Case 4: View a day's tasks

Use Case ID:	4
Use Case Name:	View a day's tasks
Created By:	Babilshia, Sreelekha, Rithika
Last Updated By:	Babilshia, Sreelekha, Rithika
Actors:	User
Description:	A user can view a day's tasks.
Preconditions:	1. User is viewing a calendar with tasks on it.
Postconditions:	1. The user can see a list of tasks for the selected day.
Normal Flow:	1.0 User views a selected day's tasks
	1. The user selects a project with existing tasks.
	2. The project calendar should display all tasks for each specific day
Alternative Flows:	None
Exceptions:	None
Priority:	High
Frequency of Use:	Common

3.7.5 Use Case 5: Switch Between Projects

The user can switch from viewing one project calendar to another.

3.7.6 Use Case 6: View Previous or Next Month

The user can switch from viewing the previous month or next month from the current month their calendar is focused on.

3.7.7 Use Case 7: Add Users to a Preexisting Project

The user can add new users to the currently selected project.

3.8 System Features

3.8.1 Project

Description and Priority
As a sub-system of a project management application, projects are an essential classification for the system and its functionality is of the highest priority. In the instance of this planner application, a project is primarily for organizing tasks. When a project is shared, all users can view and create tasks related to that project.

Note: See Section .1 for the definition of Project.
  Functional Requirements

Functional Requirement	Description
FR-1.1	Users shall be able to create a new project.
FR-1.2	When creating a project, users shall be able to provide a project name.
FR-1.3	Once a project is created, a user shall be able to view a calendar that represents the project.
FR-1.4	A user shall be able to add other users to a preexisting shareable project.
FR-1.5	A user shall have access to their own personal project.

FR-1.6	A user shall be able to cancel creation of a project.
FR-1.7	A user shall be able to switch from one project to another.

3.8.2   Tasks

Description and Priority
The priority for this system feature is high.

Note: See Section .1 for the definition of Task.

Functional Requirements

Functional Requirement	Description
FR-2.1	Users shall be able to add tasks to a project.
FR-2.2	Users shall be able provide a task title to the task.
FR-2.3	Users shall be able to assign the task to a user.
FR-2.4	Users shall be able to add a start and end date to the task.
FR-2.5	Users shall be able to add a description to the task.
FR-2.6	Users shall be able to cancel creation of the task

3.8.3   Calendar

Description and Priority
A calendar is the visual representation of a project. Tasks added to a project
will be displayed on the project calendar. As the primary interface of the
application, the priority of this system feature is high.
Functional Requirements

Functional Requirement	Description
FR-3.1	A user shall be able to view tasks related to a project.
FR-3.2	A user shall be able to navigate to a previous month or the next month.

FR-3.3	A user shall be able to view a day with tasks related to it from the project calendar.
FR-4	A user shall have all tasks assigned to them synced to their personal calendar.

3.9   External Interface Requirements

3.9.1   Hardware Interfaces

Hardware Requirement	Description
HI-1	The API shall run on any device with internet access.

3.9.2   Software Interfaces

Software Requirement	Description
SI-2	The API shall use OAuth for user authorization.

3.10   Other Nonfunctional Requirements

3.10.1   Performance Requirements

1. Projects must take no longer than one minute to update revisions.

2. Project revisions must take no longer than one minute to be processed by the server.

3. Projects revisions must take no longer than one minute to be received from the server.

4. The planner service must be online/functional 95% of the time.

3.10.2   Safety Requirements

1. Users are responsible for managing the content they view.

2. The application will not censor, modify, or restrict user language.

3.10.3   Security Requirements

1. The server must support user authentication.

4 Architecture Design

4.1 Introduction

This document graphically breaks down the software architecture for the calendar planner web application. Included is a deployment diagram, dialog map, and database entity relationship diagram. For more context on nonfunctional and functional requirements, refer to the SRS and Vision Scope sections.

4.2 Problem Description

This planner application sets out to provide a convenient feature in the solution to project and task management. Similar to other calendar and project management systems, the system will allow users to create a project, add tasks, assign tasks, add users, and display the tasks for the project. As a convenience, all tasks assigned to a user will be synced to the user's personal project/calendar.

4.3 Solution

4.3.1 Overview

The calendar application will consist of a web application available via web browser. Authentication and a realtime database will be managed through Firebase's services. The client will interact with Firebase servers to manage the creation of projects and user interactions with them. All data will be stored in Firebase's database server.

4.3.2 Components

Deployment Diagram

The Deployment Diagram consists of the client browser and the web services provided by Firebase. The application interacts with the data hosted in Firebase. Users of the application are authenticated using the Firebase's Auth Web API.
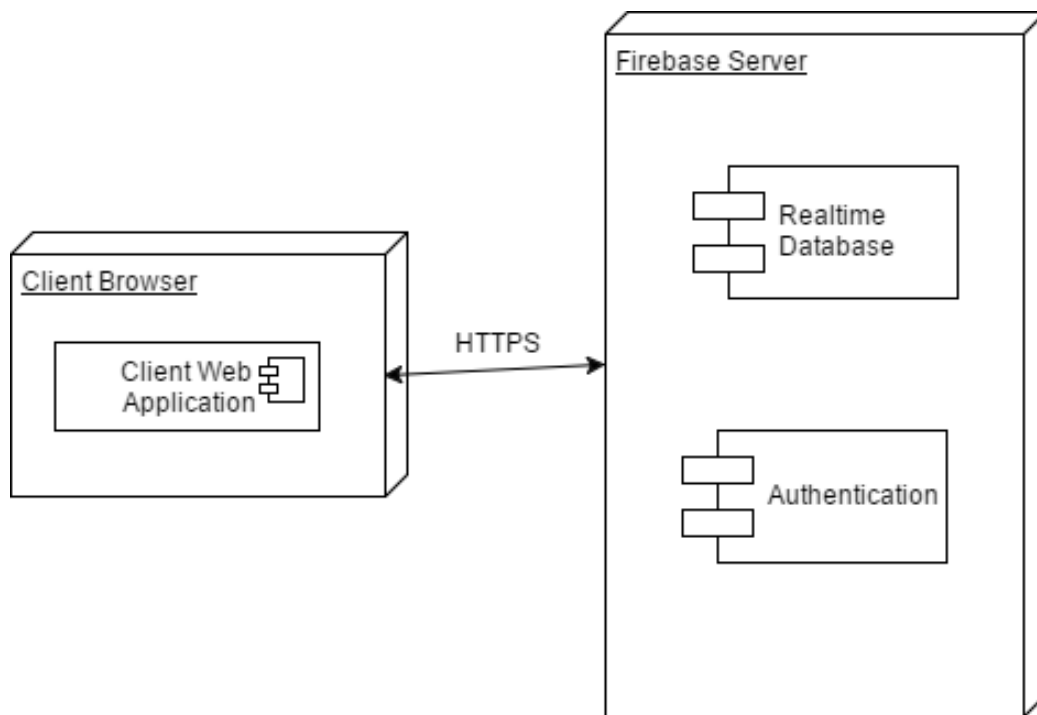
The Firebase Auth service provides the core application with a means of verifying Users and protecting User information through the use of unique, secure authorization tokens. All communication between the Auth servers and the core application is done in compliance with SSL protocol.

The Firebase services component of the system provides back-end support to the application. Firebase provides a realtime database management service that can be accessed through the use of Node.js or HTTP requests. This will allow the application to store vital information in a decentralized location for easy access. [5]

Dialog Map

The dialog map above shows the sequence of screens the user can traverse
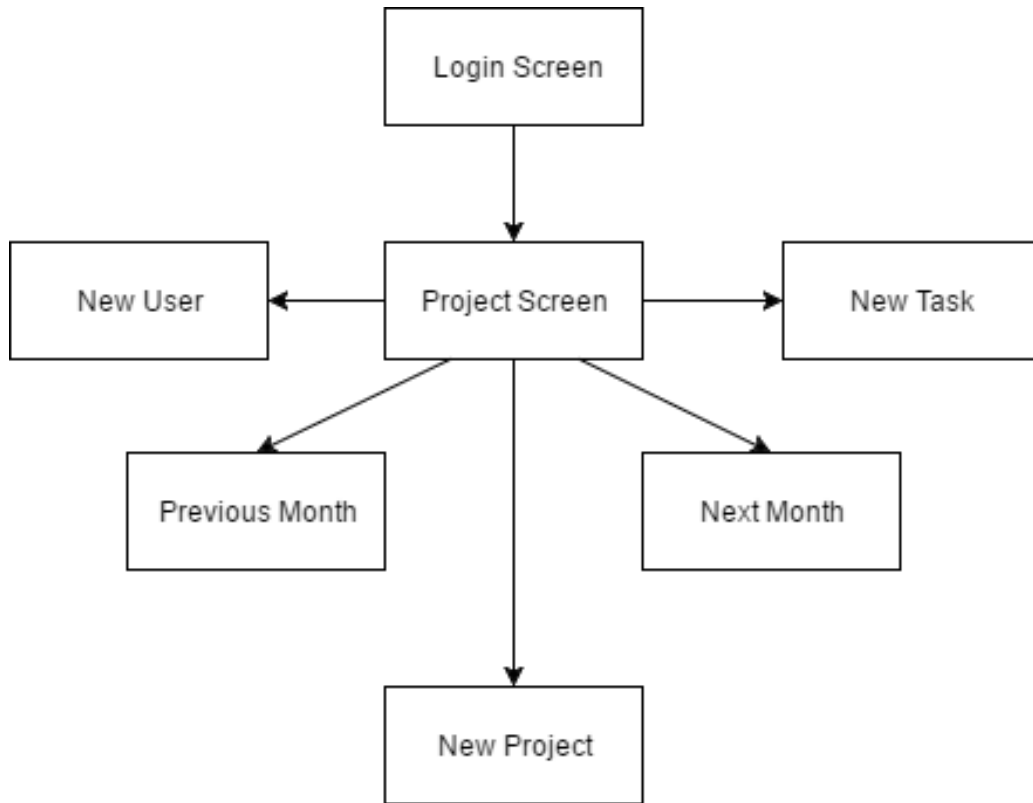
through in the planner application. Upon first use of the application, the user is directed to the login screen, where they will be prompted to login via their Google account. After creating an account, the user is sent to a screen where they view their personal project in the current month. From here the user can perform a number of actions; they can switch projects, view the previous month, next month, and create new tasks or add new users.

4.4 Design

4.4.1 Database Entity Diagram
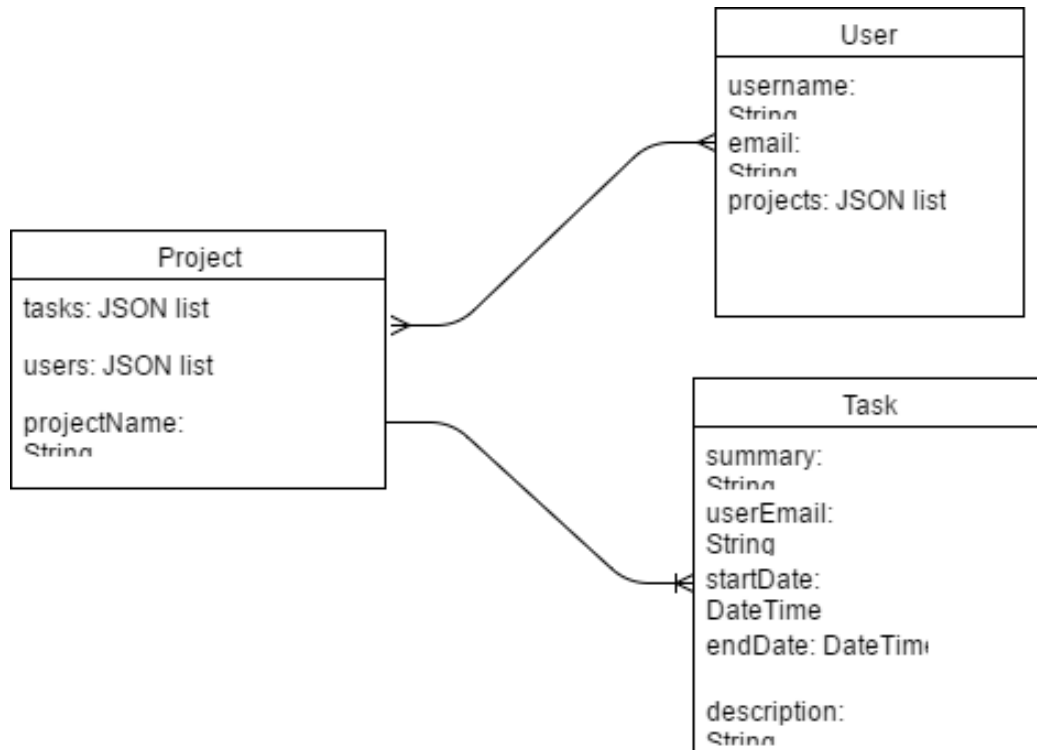
    Firebase's Realtime Database uses JSON object trees to store and organize data. For the application, the database will be organized in two main trees. One tree will be the User tree, the other will be the Project tree.

1. User

   Users that have registered to use the application and provided Google credentials for authentication are stored in the User tree. This tree is organized by the unique id of each user provided by Firebase. Under each user is three fields.

   - the user's email
   - the user's first and last name
   - a list of projects the user is a part of

2. Project

The project tree is organized by the project's unique id provided by Firebase upon the project's creation. A project in the tree contains three fields

- The name of the project
- A list of the users in the project, by user id
- A list of the tasks in the project, by task id

3. Task

Tasks will be kept track of in the Project tree, and organized by their unique task id, also provided by Firebase upon creation. Tasks will consist of five fields.

- The summary of the task, which briefly describes the task.
- The email of the user assigned to the task.
- A start date representing when the task should begin.
- An end date representing when the task is due.
- A description of the task, which describes in more detail of the task.

5 Reflection

This project served as both an excellent exercise to test my software development knowledge as well as an excellent learning experience. Web application development is a handy skill set to have as society moves more toward digital technology resources. Companies need websites and web applications can be easily reached on any device connected to the internet.

Since this was my first experience with web development technology, there was a large learning curve. My decision to use HTML5, CSS, and JavaScript to develop the application was largely from my understanding that these were the technologies generally used in web development. However, these were also technologies I already wanted to learn, which made this a more subjective than objective decision.

During the first six weeks of the project, I took a Waterfall-like approach. I developed the Vision and Scope, SRS, and Architecture Design documents. I also spent three of the six weeks experimenting with the technologies I would use, such as what development environment to use and how to develop an application using Firebase. This period allowed me to gain a strong understanding of how I wanted to develop the application. However, when it came to the actual coding, development did not go as smoothly as I expected. When I was coding, there were usually short periods of time with large amounts of productivity accompanied by longer periods of time when I would be stuck debugging a seemingly simple error. These periods of debugging took place mainly at the beginning of development. As my knowledge of JavaScript grew, debugging certain issues became relatively easy compared to the previous weeks.

I implemented the application incrementally in a more Agile fashion, by determining the order of significance for features. This also included the UI design of the application, where I left the UI design of the application until all the features were successfully implemented. This would have been a bad decision if I had ran out of time before implementing all the features, because the UI would then have consisted of the default appearance for displaying HTML files. Luckily, all the features were not overwhelmingly difficult to implement. I particularly enjoyed applying the CSS after implementing functionality. Unlike creating UI for other application platforms, CSS was easier and felt more customizable.

Overall, I enjoyed developing this application. Although the first few weeks were filled with more documentation writing, I believe this was a good

decision because it allowed me to gain a strong understanding of how the application would work and should be developed.

Glossary

Term	Definition
Project	A project is defined as a collection of tasks related to a common topic.
Task	A task represents a work item in a user's schedule.

Issues List

No known issues.

References

[1] Asana. Asana, 2017.

[2] Google Firebase. Firebase, 2017.

[3] Google. Google calendar, 2017.

[4] Jennifer Greene. Building better software, 2015.

[5] Anant Narayanan. Where does firebase fit in your app, 2015.

[6] Margaret Rouse. software requirements specification, 2007.

[7] Aspguy's Weblog. What is a software architecture document, 2011.