Resource Planner:
A Progressive Web Application

***Abstract*—Abstract—The Resource Planner Progressive Web App is a system designed to automate the process of creating and delegating tasks by assigning them to users and implement to work as both a desktop and mobile app with offline support. Index Terms—PWA, Progressive Web Application**

***Index Terms*—PWA, Progressive Web Application**

I. Introduction

A resource planning software or a resource management software is a tool used to manage resources or people and schedule of a company or an organization. It helps in allocating or delegating tasks to people where it is needed. Some of the main features of a resource planning software includes Gantt charts, timesheets, workload management, alerts and real-time data, reports and portfolio management. [1]

With smartphones becoming a vital part of day-to-day existence in the modern world, apps are quickly replacing the browser in becoming the default medium used in interacting with websites. Though traditionally requiring the costly and time-consuming development of separate systems using different languages and different technologies for different platforms, the rise of Progressive Web Applications (PWA) offer to become a solution to the cross-platform complexity and difficulties of maintaining separate apps for different platforms.

In a nutshell, PWAs are basically just websites which behave like native applications on mobile devices. This eliminates the need for the separate apps for different platforms through the existence of a single codebase - which is just the main website. This also ensures that the version being used by all users is the latest available version, removing the need for updating native apps through app stores.

One major difference between websites and native apps is the ability to continue interacting with the app even when the device is offline. In traditional websites, users are not able to continue browsing the site without internet connection unlike native apps which are able to store preloaded data for offline use. To mimic this behavior, PWAs take advantage of Service Workers. This allows the website to run offline by storing the last loaded data in the browser cache. This also speeds up subsequent loads by using cached data, allowing users with slow or intermittent internet connections to have a faster user experience when interacting with the app.

The objective of this project is to develop a resource planning PWA where managers can create, review and assign tasks to developers in order to better manage the allocation of resources.

A. Objectives

The general objective of the project is to develop a system for the creation and management of tasks and users. The specific objectives of the project follow:

- User Management
  - To create a system for creating, retrieving, updating and deleting of users.
- Task Management
  - To create a system for creating, retrieving, updating and deleting of tasks.
- Offline Support
  - To implement persistence of data and handling of some functionalities even when the site is offline.

B. Review of Related Literature

There is a number resource planning software available today. Some of notable examples are Teamdeck, Float, and Forecast.

Teamdeck is a simple resource management tool without advanced features. It has a simple user interface and low learning curve suitable for tight budget companies or organizations. Some of the key features of Teamdeck are as follows: Notifications when delegation overlaps, leave management for resource availability, expertise-based assignment, and sync with other project management apps.

Float helps visualize team availability. It also has a real-time notification when changes are made keeping the whole team updated. [2]

Forecast is a cloud-based project management tool. It helps keep track of project requirements. It also has a feature to predict requirements ahead of time. Some of its features also include Resource scheduling, time-tracking, multiple user collaboration, alert, and notifications. [3]

PWA stands for progressive web app. This is an app built from the web technologies we all know and love, like HTML, CSS, and JavaScript, but with a feel and functionality that rivals an actual native app. [4] With relatively minor modifications, almost all websites can be quickly turned into progressive web apps. This makes developing and maintaining apps more convenient and cost-effective thanks to only having to maintain a singular cross-platform codebase, compared to developing separate apps for the different mobile platforms.

PWAs have long been gaining popularity in the tech industry. Big tech companies such as Twitter, Pinterest and Uber are some who implement PWAs. Twitter Lite, a PWA developed by Twitter, became the default mobile app for all

users globally since April 2017, and has led to 65% increase in pages per session, 75% increase in tweets sent and 20% decrease in bounce rate. Nicolas Gallagher, the Engineering Lead for Twitter, notes that "Twitter Lite is now the fastest, least expensive, and most reliable way to use Twitter. The web app rivals the performance of our native apps but requires less than 3% of the device storage space compared to Twitter for Android." [5]

Service workers are one of the main components in developing a PWA and is used to store loaded in the cache for subsequent use. First loads for Twitter Lite for example clock in at under 5 seconds over 3G networks on most devices, and once stored by a service worker, loads almost instantly on subsequent loads even on flaky networks.[2] With mobile users accounting for over 80% of Twitter's usage and with many Twitter Lite users recorded to be using the site via 2G and 3G networks, a fast user experience even in slower connections is essential in maintaining user engagement. Repeat visits are nearly instant thanks to service worker caching of current views, feed updates, notifications, messages and settings. Twitter Lite users experience a 50% reduction in 99th percentile time-to-interactive latency and logged in users have a 30% reduction in average load time. [5]

II. Methodology

A. Functional Requirements

The following are the detailed functional requirements of the system:

- Administrators can perform the following:
  - create, view, update, delete users
  - create, view, update, delete tasks
  - approve user registration
- Project Managers can perform the following:
  - create tasks
  - view, update and delete tasks they created
- Developers can perform the following:
  - retrieve and update tasks they have been assigned to
- Task management module must be functional even when site is offline
- Users without accounts may register an account which must be approved by an administrator
- Users may request for new passwords in case they have forgotten their passwords
- All users may change their passwords
- Reports page can be accessed by the administrator only. The reports consist of Time Allocation per Developer, Number of Tickets by Priority and Number of Tickets by Effort presented in a bar graph. The reports can be filtered with date range

B. Modules

- Log In Module
  - The system will have three types of users with varying levels of access: Administrators, Project Managers and Developers.

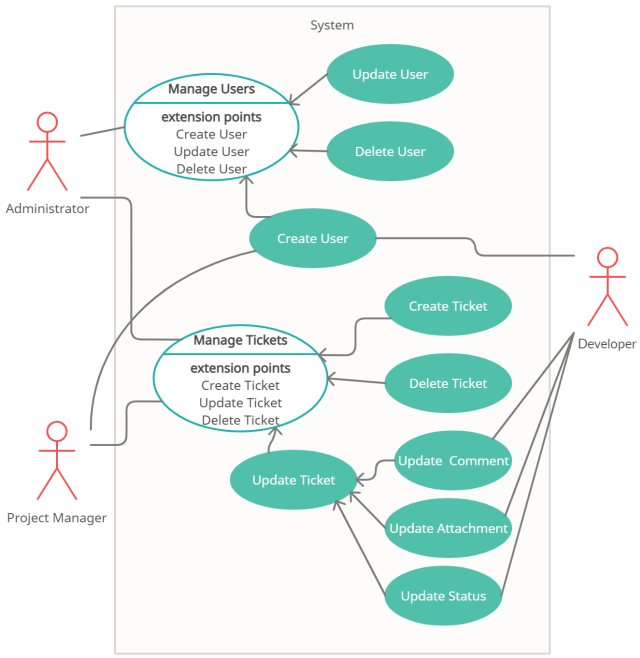

Fig. 1. Resource Planner use case diagram

- User Management Module
  - This module may only be accessed by administrators. Administrator will have full read and write access over all user accounts. Administrators will also be able to approve account registrations here.
- Task Management Module
  - This module may be accessed by all roles. Read and write access to data will depend on their level of authentication. Administrators will have full read and write access over all tasks, Project Managers will only be able to have read and write access over tasks which they themselves have created and developers will only have read access over tasks which have been assigned to them. Developers can also update the effort, status, attachment, and discussion sections of the task they are assigned in. This module will be implemented to be usable even when offline.
- Registration Module
  - Users without accounts may register for an account by filling out a form with all necessary information and then filing a request. Administrators will have to approve the account registration request before the account can be used to log into the system. An email will be sent notifying the user once their registration has been approved.
- Forgot Password Module
  - Users may request for new passwords in case they have forgotten their passwords. An auto-generated password will be sent to their email which they can they use to log into their accounts.

- Change Password Module
  - All authenticated users may change their passwords.
- Reports Module
  - Administrators can access this module. The reports are presented in bar graph filtered with date range. The reports consist of the following: Time Allocation per Developer, Number of Tickets by Priority and Number of Tickets by Effort.

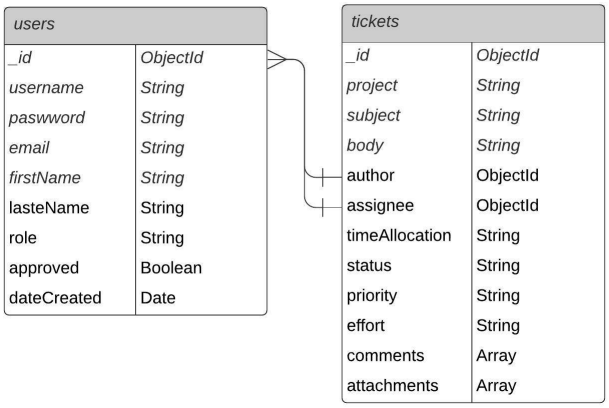

Fig. 2. Resource Planner entity relationship diagram

III. Results and Discussion

The web application was developed using React and Redux for the client, Node and Express for the server and MongoDB for the database. Offline persistence of static assets was implemented using Service Workers and Redux-Persist for preserving Redux state. Handling of http requests when offline was implemented using Redux-Optimist and AxiosRetry. Offline handling of http requests was implemented by declaring a set number of retries and then updating the system according to result.

After successfully logging in, the user is directed to the user dashboard. The dashboard contains the Task Management Module. Here users may view, create, update and delete tasks. Permissions will depend upon the user's role. Administrators will have access to all functions and all tasks, Managers will be able to create new tasks and view, update and delete tasks which they have created, and Developers will be able to view tasks assigned to them but are not able to create or modify these tasks. Developers may post in the discussion sections, update effort, update status and add attachment of the tickets they are assigned in.

For easier browsing of tasks, users may filter the displayed list of tasks by project, developer and or priority.

If the user is an administrator, a link to the User Management Module will be available in the navigation bar. Here administrators may create, view, update and delete users. Users without accounts may register a new account by following a link from the Login page to the Registration page and then

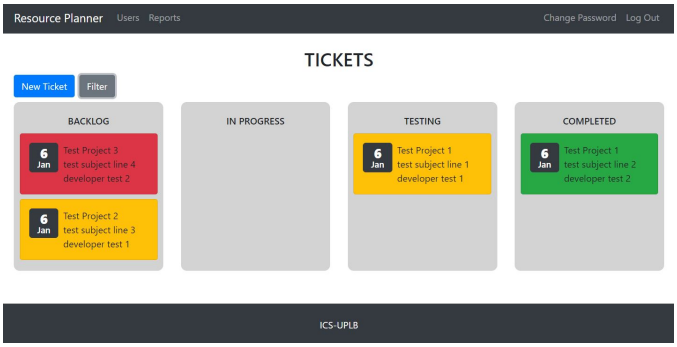

Fig. 3. Task Management Module

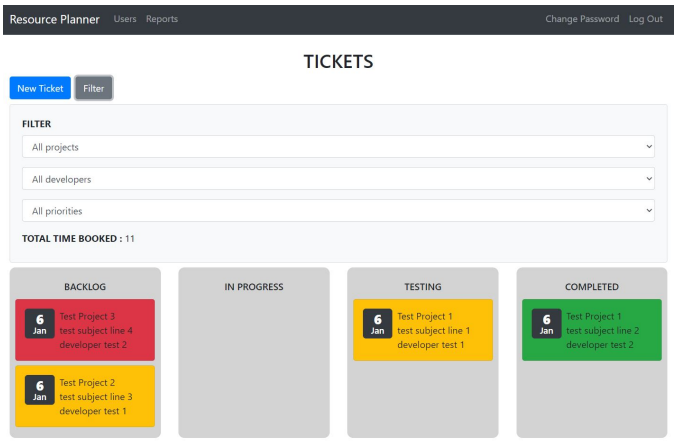

Fig. 4. Filter by Project / Developer / Priority

accomplishing the registration form . An administrator must then approve the registration before the account can be used to log into the system. Administrators may also view and approve new account registrations here. New account users will receive an email notifying them once their registration has been approved.

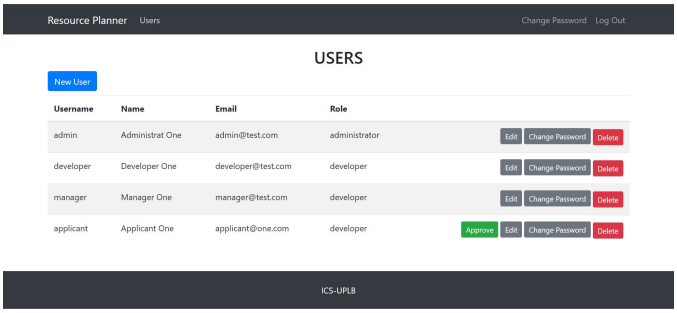

Fig. 5. User Management Module

Service workers are implemented to allow users to continue interacting with the site even when they are offline. Users may continue browsing tasks even without internet connection and data will not be lost and can be reloaded even after closing the browser.

Updating and deleting of tasks are also implemented to be usable even when offline using a http retry system. Implemented using the axios-retry library, number of attempts and duration of delay between attempts can easily be modified. When the number of attempts runs out, users will be notified of the client failing to communicate with the server. If successful, the user interface is updated to reflect the changes in the data.

```
axiosRetry(axios, {retries: 20,
retryDelay: retryCount => {return
retryCount * 1000}});
```

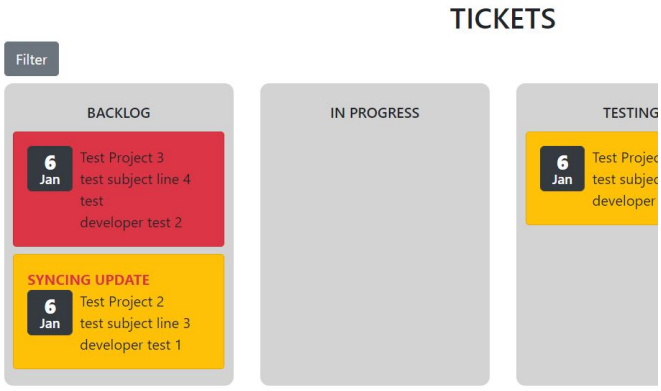

Fig. 6. Update task while offline

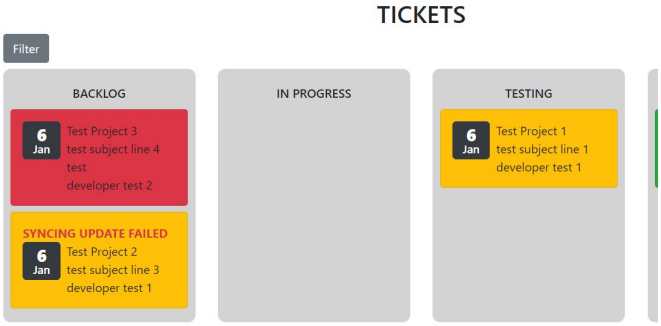

Fig. 7. Number of attempts exceeded

We have conducted a System Usability Scale (SUS) survey from several software-related personnel. Most of the respondents works or have worked with my previous company, Liquid Interactive. The SUS is a tool for measuring the usability of a system. It consists of 10 item questionnaires with 5 options ranging from Strongly Disagree to Strongly Agree. We have gathered 10 respondents to review the Resource Planner application. The respondents are provided with a brief user guide for the system.

Our new system has received a total score of 85.75. This score can be further interpreted as having an excellent usability scale with a ranking of B.

Scales			
Odd items	Even items	SUS score (/100)	Grades
17	19	90	A
20	20	100	A
20	9	72.5	C
10	18	70	C
20	20	100	A
20	19	97.5	A
20	20	100	A
17	17	85	B
16	10	65	D
16	15	77.5	C
	Average score	85.75	B

Grading SUS Key	
92	Best imaginable
85	Excellent
72	Good
52	OK/Fair
38	Poor
25	Worst imaginable
90-100	A
80-89	B
70-79	C
60-69	D
Less than 60	F

Fig. 8. Resource Planner SUS score

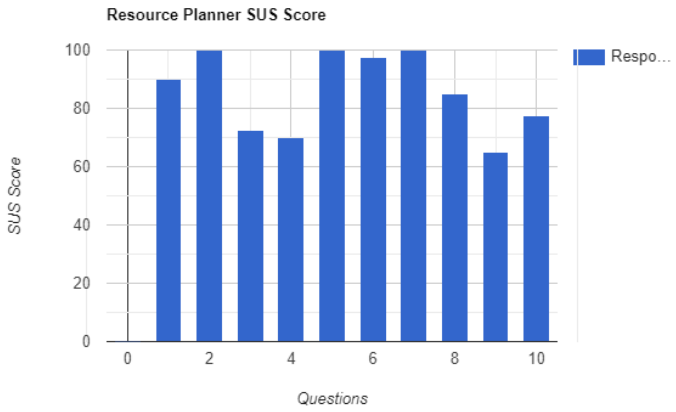

Fig. 9. Resource Planner SUS score with grades

IV. Acknowledgment

I would like to express my deep and sincere gratitude to my advisor, Joseph Anthony C. Hermocilla, Assistance Professor, Institute of Computer Science, College of Arts and Sciences, University of the Philippines Los Baños, for giving me the opportunity to finish my research and for providing invaluable guidance throughout this research. It was a great privilege and honor to study under his guidance.

I am extremely grateful to my parents for their support, love and sacrifices for my future. I extend my thanks to my family for their support and for the comfort they always provide. This research would have not been possible without your unending support and guidance. My special thanks to Liquid Interactive Team specially to John Andrew Palaganas and Jocelyn Cardenas for giving me invaluable insights and suggestions with my research as well as handling the communication with the team.

References

[1] https://www.projectmanager.com/resource-management
[2] https://teamdeck.io/project-management/top-resource-management-tools/
[3] https://www.eresourcescheduler.com/blog/best-resource-management-softwares-and-tools
[4] https://yoast.com/what-is-a-progressive-web-app-pwa/
[5] https://developers.google.com/web/showcase/2017/twitter