only a few lines of JavaScript and HTML. For assistance categorizing transactions several APIs are being considered including Google Shopping. The marketing data for an item can provide valuable information on how it will be utilized by the customer and how necessary the item is to their budget. Utilization will depend on what specific algorithm is selected for categorizing an expense as a luxury or necessity and how much of the data processing is conducted within the native application. For the data storage Mongo DB will hold user statistics and budgets. Mongo DB is a free and open source database system which is considered NoSQL and relies on JSON and documents instead of tables for its structure. The system architecture is shown below in Figure 14.

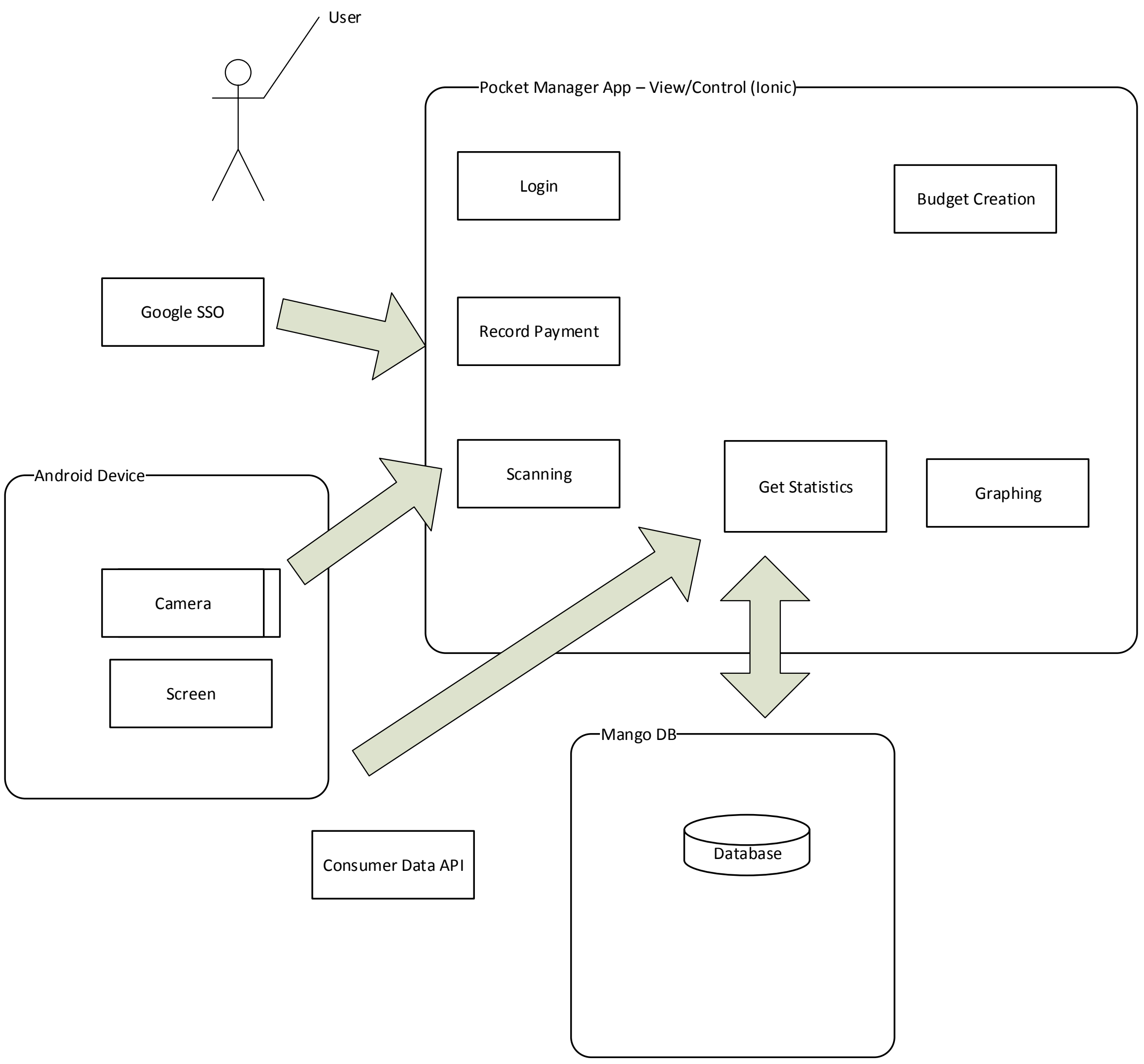


Figure 14. System architectural designation of the *Pocket Manager* to be developed.

Figure 15 and Figure 16 are the high-level UML class diagram and sequence diagram, respectively.

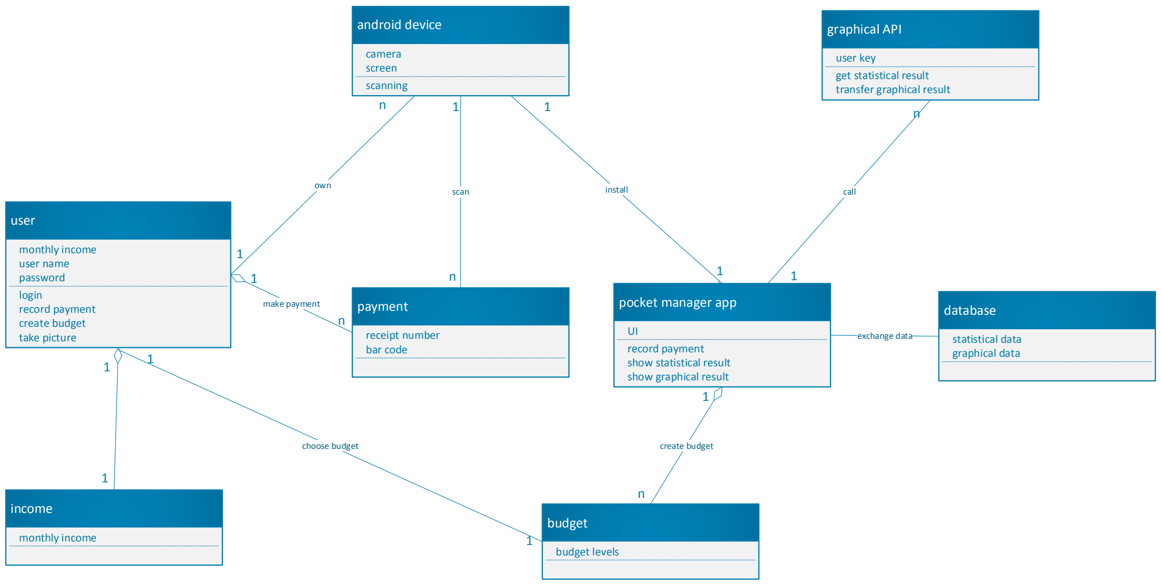


Figure 15. High-level UML class diagram for *Pocket Manager* system designation.

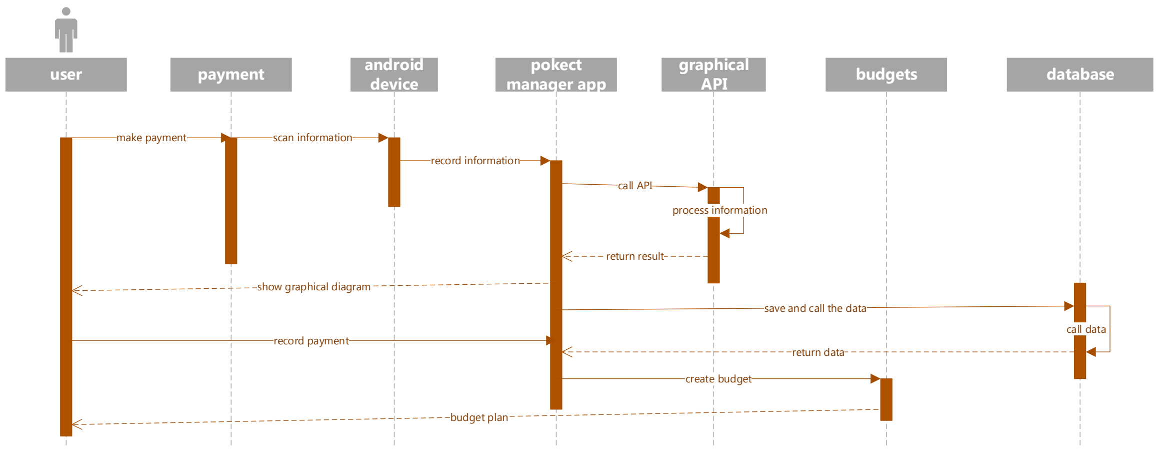


Figure 16. High-level UML sequence diagram for *Pocket Manager* system designation.

- **Testing**

1. Overview

Testing of the application involved a multi-step process including pre-submission audit, virtual device testing, and physical device testing. The process was designed to find problems in both the design of the code and features of the application.

2. Coding Audit

The following checklist is run on code at the start of testing in order to find errors early in the process and make it easy to find the root cause of a bug. The list must be completed by a group member before the coding submission is considered complete.

For the pre-submission audit, several checklists were developed to check the quality of code. The first section deals with the commenting and readability of the code. Code with clear formatting allows for easier troubleshooting and makes it easier for other users to modify code. For outside auditing and grading, clear formatting is a critical part of making it possible for outside parties to provide feedback. As a result, checking the format and documentation of code is a significant section of testing process.

If any software bugs or potential enhancements were uncovered before the submission, a description was posted as an issue. The team leader could determine how to resolve the issue by using any of the following options including accept the risk, assign someone to fix the issue, or seek outside help. Table 17 is the result of the coding audit for our system.

Table 17. Coding audit checklist for the *Pocket Manager* system.

Item	**Question/Comments**	**Response**
a.	Does every section have at least one comment? (about one comment for every four lines of Java/JavaScript or one for every major section of html)	HTML could use a few extra comments to help it stay organized in the future
b.	Is the code neat? (not too many blank lines)	Yes
c.	Is proper spelling and grammar used?	Yes

Item	**Code**	**Response**
d.	Is the proper indentation used?	Yes
e.	Are variable and function names meaningful?	Yes
f.	Is major functionality subdivided logically into classes and activities?	Yes
g.	Is camel case used for functions and variables?	N/A

Item	**Feedback**	**Response**
a.	Were major issues submitted to GitHub?	No major issues
b.	Do you have any major concerns about your code?	Not yet

3. Virtual Device Test

At least two versions of devices are used in the virtual testing including the following

a) Phone - Nexus 5x with 5.2in screen and API Level 23
b) Tablet - Nexus 9 with 8.83in screen and API Level 23
c) Wear – Android Wear Square with 1.65in screen and API Level 23

As part of the Virtual Testing the following checklist is used to make sure all of the features are fully tested. The list must be completed by a group member before the coding submission is considered complete.

The checklist has two major sections. The first part covers the user interface or the look and feel of the application. The detailed evaluation is designed to insure that the basic rules of good user interface are designed. Since developers may become highly engrossed is their work, certain obvious problems may be missed due to familiarity with the work. As a result, the below checklist forces developers to check each other's work in detail in order to catch mistakes before the software is deployed and in user hands.

The next major section involves the functionality of the application. Any new feature or modified area should be tested. In addition, the existing features should be tested to confirm that changes did not negatively affect existing functionality. Finally, the transition between major device states is tested. For example, the app can be exited and restarted to confirm no functionality is lost when reopened. In addition, the device can be restarted to confirm that the application still functions. See Table 18 for the details of our analytical results.

Table 18. Virtual device checklist.

Item	User Interface	Response
a.	Visibility – Are all parts of the initial page visible?	Yes, all are visible
b.	Alignment – Does the alignments of parts look correct?	Yes
c.	Color – Do the colors appear as expected?	Yes
d.	Screen Changes – Does the page look correct from both screen orientations?	This will be more of an issue in future increments when more buttons are on the screen
e.	Keyboard – Can the app features still be used if the virtual keyboard pops up?	Same as above

Item	Features	Response
f.	Do all new features work as expected?	Yes
g.	Do existing features still work?	N.A

Item	Transitions	Response
i.	Does the app work if someone exits then opens the app again?	N/A
j.	Does the phone function is the app has been running for over five minutes? (test of memory leeks)	N/A

4. Physical Testing and User Feedback

Physical testing involves adding the app to a physical device, showing potential users, collecting feedback to utilize in future increments.

Currently, we are searching for physical devices for application install. Some group members use iPhones and others have older versions of Android OS. Several tablets rom Dr. Song's lab may be used, but they may not be a valid test since they are rooted for other experiments. As a result, the following feedback was from virtual devices until we can secure the proper physical devices. With the large variety of devices in the android ecosystem, the long-term goal is to secure multiple types of devices for testing.

The user feedback was obtained from three people with a variety of technical experience. The goal was to obtain objective feedback on the application which caught errors missed by the previous checklists and provided ideas for new features and enhancements.

Due to the simplicity of the first increment the user feedback was combined on a single form. In future increments, the amount of feedback is expected to expand. As a result, user feedback forms will be individually reported with a brief summary in the increment documentation (see Table 19).

Table 19. User feedback questions and responses.

Item	Question	Responses
a.	What do you like about the app?	Simple looks nice
b.	What do you dislike?	Id numbers were confusing, did not look very flashy
c.	Any suggested changes?	Add a nicer background

5. Going Forward

As the software increases in complexity, testing is expected to be a much more important part of the development process. The checklists developed are expected to grow and be utilized by more than one team member. TA Feedback on the process will be a critical component in what direction the testing takes.

User feedback for testing will become a means of marketing and deployment. Future increments are expected to develop tools to monitor behavior by the users testing the application in order to have mathematical data on how the application is actually utilized. In addition, a support request and feedback button is expected to be added to give the user a way to quickly send feedback to a shared email account. Having a high number of outside testers provides a way to both develop marketing material and an existing set of users before the official application release to the market.

- **Deployment**

1. Overview

Deployment includes leading the app onto devices, taking screenshots with detailed descriptions, and creating a wiki in GitHub. The overall purpose of the section of the process is to deliver the completed application to market (or in the case of the first increment, the teachers). With the fast pace of technology and the high cost associated with employing highly skilled developers, delivering a product early is critical for a technology startup. Deployment moves the application from the developer world to the real world of users. Both data and revenue from customers can be obtained providing critical fuel for future increments.

The deployment of Pocket Manager’s initial increment focused on the delivery of the skeleton of our app including the user interface and development process. The user interface was developed in ionic with the main objective being to make sure our development tools worked properly to create a hybrid application. In the case of some group members the development environment alone took several days to get up and running. With the successful setup of ionic and the development of the core pages, a solid foundation was built for future increments. Since the look and feel of an application creates a first impression with the user critical to marketing, the deployment of the interface design was intended to get feedback early and thereby improve the interface early in the process. Overall, changing the interface early in the process is preferable in order not to confuse the user by a modifying where items are found in the application. As a result, the initial deployment focused on CSS, HTML and other view components.

Along with the interface, the initial increment's deployment focused on refining our processes for development and release of the application. Along with a basic structure for the app in GitHub and Ionic, the process of submitting, testing, and deploying the software was a key deliverable for the initial increment.

The major concern with the deployment is the limited number of compatible devices available for testing the application. Due to the ever growing variety or Android devices, a large number of devices would be preferred for initial deployment. Since our application is not on the app market yet, the team relies on devices available through people known by members to deploy and test the devices.

2. Screenshots

Below are the primary screenshots for the application along with the detailed descriptions.

a. Login Screen (Figure 20)

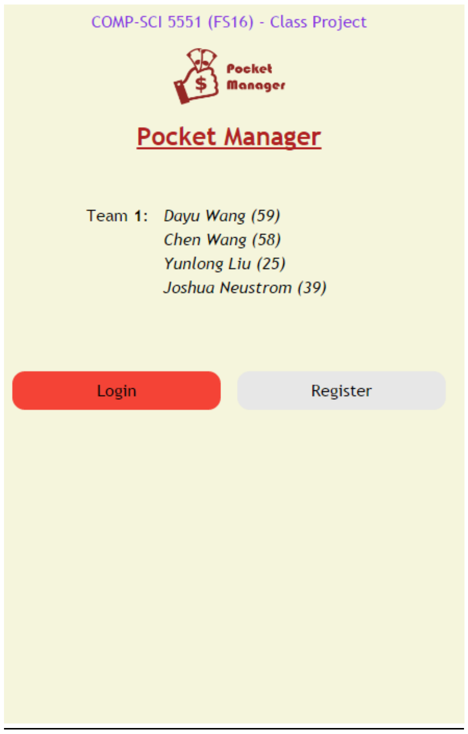


Figure 20. Login screen.

The initial login screen provides the users first interaction with the application. A darker tan colored background was selected to be easier on user eyesight while still allowing for ample contrasts for users with color blindness.

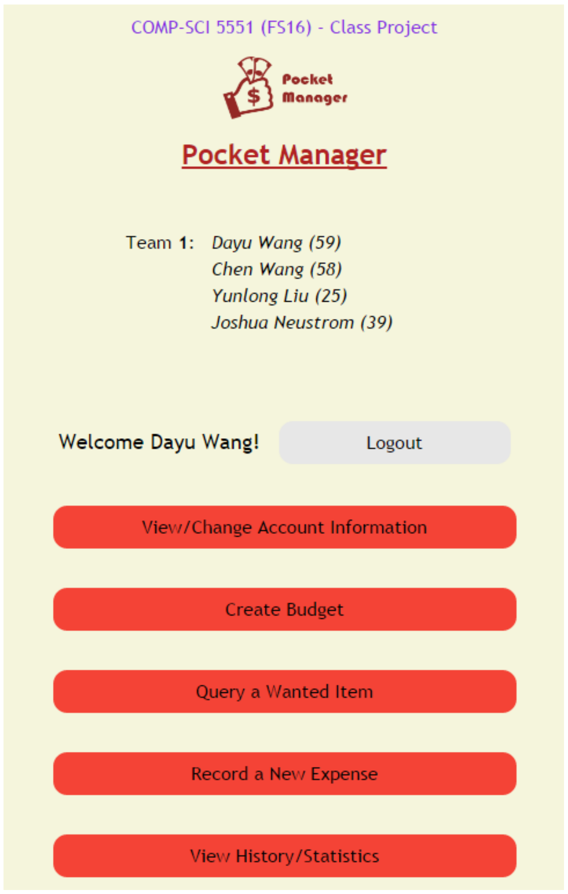
The two primary buttons are for login and registration of the user. The buttons are designed to be responsive to user touch or the hovering of a mouse by having two side arrows appear. The buttons also have unique colors for easier differentiation by users. The login button allows users to access the main page with all of the key functionality and confirms that the user is properly authenticated. The registration button is intended to take users to the registration page to collect key user information including preferred credentials for the application database.

For branding purposes, the Pocket Manager logo is central to the page creating a visual association in the users mind. The logo also allows for a user to know if they are using the real Pocket Manager Application.

At the head of the page is the basic group information. Although such information would be unlikely to be included in a real world app in the Android market, the title was included to help the graders and teachers easily identify the project and members.

Going forward, new features are expected to be added such as request help and Google Single Sign On buttons.

b. Main Menu (Figure 21)


COMP-SCI 5551 (FS16) - Class Project

Pocket Manager

Team 1: *Dayu Wang (59)*
Chen Wang (58)
Yunlong Liu (25)
Joshua Neustrom (39)

Welcome Dayu Wang!

Figure 21. Main menu screen.

The main menu screen provides the user with access to all of the main features of the application and serves as a one stop shop for the user. At the top of the list is a logout button to take the user back to the login page and revoke their session access to the main menu. Features accessed include the option for changing account information initially registered by the user including the important function of updating the password. The next button is for budget creation which allows a user to create a budget for spending including the top level categories of transactions and the max spending each area. Further down, a button exists for query of an item to see if the item is a luxury or necessity. The function is the key advantage of the Pocket Manager which allows a user to cut spending by having an outside source label the potential purchase as a luxury if it is not truly needed for the budget. Continuing down the list, the screen has a button for adding actual expenses in the context of what was originally budgeted. The bottom button takes the user to a big picture view of their transaction history to allow them to understand their spending habits and success in meeting the budget over the long term.

The overall look and feel from the initial login page was maintained in the main menu section to create a consistent interface for the user. Buttons respond in a similar manner with two arrows appearing when touched or hovered over by the user. Furthermore, the key project information is kept on the title section of the page.

3. Wiki Page

A Wiki page was created for the increment 1 which recursively includes the report. The page allows an outside user to easily understand the deliverable included in our Github source folder. In addition, the GitHub contains a source folder for all finalized code that was part of the deliverables. Furthermore, the main screenshots from the report and a copy off the report istself can also be found in the documentation folder on the GitHub site. Finally, a readme exists in the core section of our repository with the key links and the overall project info.

- Wiki- https://github.com/dwk894/CS5551FS16_Pocket_Manager/wiki/Increment-1
- Documentation- https://github.com/dwk894/CS5551FS16_Pocket_Manager/tree/master/Documentation
- Source Code - https://github.com/dwk894/CS5551FS16_Pocket_Manager/tree/master/Source
- Readme- https://github.com/dwk894/CS5551FS16_Pocket_Manager/blob/master/README.md

4. Going Forward

Future increments are expected to build on the structure created in the initial increment. Each increment will have a separate Wiki page and the overall source and documentation structure will be maintained.

- **Bibliography**

[1] Project on Student Debt, institution: The Institute for College Access & Success
http://ticas.org/posd/map-state-data-2015
[2] The Take Charge America Institute
https://tcainstitute.org
[3] https://cloud.google.com/vision
[4] https://docs.mongodb.com/manual
[5] http://www.highcharts.com/demo
[6] http://www.nyu.edu/classes/jcf/g22.3033-007/slides/session2/g22_3033_011_c23.pdf