**FP Cold-app\***

**Workbook**

\*Title is still on working progress

1. [Description](#_izb1kk1bkdqz)
2. [Functional Specifications](#_dz8kf7iivtel)
3. [User Types & Use Cases](#_nelqbv6pxc6y)
4. [Data Objects](#_kpd6q36lzxpo)
5. [User Stories](#_fcgnzq965x3v)
6. [Scrum](#_g4dit1tolnkg)
7. [Backlog](#_yt06xwomwt89)
8. [Changelog](#_fdmmiscj5its)
9. [Glossary](#_gto0ytg6qf42)

# I. Description

**FP Cold-app\*** will be a mobile application with the purpose of helping the company *First Produce: Hi-Line Cold Storage* with the storage of their products coming in and out of their warehouse. The mobile application will use a Heroku database and Parse to save and manage barcode numbers corresponding to different product’s pallet tag numbers and pre-defined location numbers which will be associated with each other in order to create the physical storage on the server. It will use an user-account system to discriminate between multiple tiers of users and their privileges within the app.

Git on GitHub will be used as version control, and new branches will be created every time a ‘big’ implementation or bug fixing takes place; all small changes may be done either on existing branches or directly on the master branch.

This application will read the barcodes in one of two ways: either by using a physical barcode scanner directly connected to the mobile phone (either by hardware or bluetooth), or by using the phone’s rear camera and the integration of the *ZXing Team* Barcode Scanner mobile app in order to read the desired barcodes.

Furthermore, this application will give certain tools to the administrators in order to manage personal and inventory, and also some inventory query systems for most if not all of the user. It will also have a transaction-based activity that will allow administrators to keep track of any use from all of the users.

# II. Functional Specifications

* Login screen is used for logging in with an unique 4-digit PIN number
* Main Menu ‘IN’ button is used to add a pallet number into a pre-defined location within the storage
* Main Menu ‘OUT’ button is used to take out a stored product from the storage
* Main Menu ‘TRANSFER’ button is used to move a product from one location to another
* Main Menu ‘INVENTORY’ button
* Main Menu ‘SETTINGS’ button
* There will be announcements at the bottom of the Main Menu screen for the workers
* A Parse-Heroku database will be used for the virtual storage
* Inventory reports will be created with multiple filters
* A transaction system will be used to keep track of the use within the app including but not limited to storing a product, taking a product out of storage, transferring a product, looking at inventory reports etc.

# III. User Types and User Cases

User Types (by Tier):

1. Administrator
2. Worker

User Cases:

Login:

The function where the app will verify the existence of a user based on a 4-digit PIN through Heroku and Parse.

Inbound:

The function where any verified user can add products into the virtual storage given a valid pallet tag and an existing location number.

Outbound:

The function where any verified user can take products out of the virtual storage given the correct pallet tag and bounded location number.

Transfer:

The function where any verified user can change the location of a product given a pallet tag already stored in the virtual storage and an existing location number.

# IV. Data Objects

User:

ID (10 digits) -- int

Tier (digit) -- int

Name (text) -- String : name of the user

Username (4 digits) -- String : same as password

Password (4 digits) -- String : 4 digit PIN to access application contents

Storage:

ID (10 digits) -- int

rackNumber (text) -- String : rack location number

Content (text list) -- String Array : contents of the location

# V. User Stories

1. *Login*:

After opening the app, the user will see a text field asking for a 4-digit PIN in order to login. The user will fill in their information and click on the ‘Login’ button in order to be taken to the main menu screen.

1. *Inbound*:

In the main menu, the user will click on the ‘IN’ button in order to go into the inbound screen, where he will see instructions that ask him to first scan the pallet tag number, followed by the location number where the pallet will be stored both in the virtual and physical storage.

1. *Outbound*:

In the main menu, the user will click on the ‘OUT’ button in order to go into the outbound screen, where he will see instructions that ask him to first scan the location number, followed by the pallet tag number that will be taken out of both the virtual and physical storage.

1. *Transfer*:

In the main menu, the user will click on the ‘TRANSFER’ button in order to go into the transfer screen, where he will see instructions that ask him to first scan the pallet tag number, followed by the new location number where the product will now be stored both in the virtual and physical storage.

1. *Inventory*:

In the main menu, the user will click on the ‘INVENTORY’ button in order to go into the inventory screen.

1. *Settings*:

In the main menu, the user will click on the ‘SETTINGS’ button in order to go into the settings menu.

# VI. Scrum

Key:

Current Sprint - Which sprint was this job done at

Job^ - Part of the application that needs to be worked on

^Indention implies hierarchy

Priority - How important it is for the overall application : 1-5

Status - N: Not started

W: Working on it

F: Finished

TA (Time Attempted)\* - Time projected for the completion of this job

TC (Time Completed)\* - Actual time that it took to complete this job

\*D/H.F : days/hours.fraction

\*1 day = 8 hours

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Current Sprint** | **Job** | **Priority** | **Status** | **TA** | **TC** |
| 1 | Setup Heroku-Parse database | 1 | F | 0/1.5 | 0/2 |
| 1 | MainActivity | 1 | W | --- | --- |
| 1 | Activity creation | 1 | F | 0/0.5 | 0/0.5 |
| 1 | UI aesthetics | 3 | F | 0/1 | 0/1 |
| 1 | UI behavior | 4 | F | 0/0.3 | 0/0.3 |
| 1 | UI layout | 4 | W | 0/1 |  |
| 1 | Title | 3 | F | 0/1 | 0/1 |
| 1 | EditText for PIN-number | 2 | F | 0/1 | 0/1.5 |
| 1 | ‘LOGIN’ Button | 2 | F | 0/0.5 | 0/0.5 |
| 1 | Button actions & behavior | 2 | F | 0/1 | 0/1.5 |
|  | Keyboard actions | 5 | N | 0/0.3 |  |
| 1 | Keyboard behavior | 3 | F | 0/0.2 | 0/0.2 |
| 1 | HomeActivity | 1 | W | --- | --- |
| 1 | Activity creation | 1 | F | 0/0.5 | 0/0.3 |
| 1 | UI layout | 2 | F | 0/0.5 | 0/0.5 |
| 1 | UI aesthetics | 3 | F | 0/0.5 | 0/1.5 |
| 1 | ActionBar setup | 5 | F | 0/0.3 | 0/0.3 |
| 1 | Welcome message + time | 5 | F | 0/0.2 | 0/0.2 |
| 1 | Announcement creation | 4 | F | 0/0.2 | 0/0.2 |
|  | Announcement functionality | 5 | N | 0/1 |  |
|  | Announcement animation | 5 | N | 0/1 |  |
| 1 | Button creation | 2 | F | 0/0.2 | 0/0.2 |
| 1 | Button functionality | 2 | W | 0/2 |  |
| 1 | InOutActivity | 1 | W | --- | --- |
| 1 | Activity creation | 1 | F | 0/2 | 0/3.5 |
| 1 | Adaptation for *Inbound* | 1 | F | 0/0.5 | 0/0.5 |
| 1 | Adaptation for *Outbound* | 1 | F | 0/2 | 0/3 |
| 1 | UI aesthetics | 3 | F | 0/0.5 | 0/0.5 |
| 1 | UI behaviour | 4 | F | 0/0.2 | 0/0.2 |
| 1 | ActionBar setup | 5 | F | 0/0.3 | 0/0.5 |
|  | ActionBar behaviour | 4 | N | 0/0.5 |  |
| 1 | Instructions TextView | 5 | F | 0/0.2 | 0/0.2 |
| 1 | Camera ImageButtons (CIBs) | 3 | F | 0/0.5 | 0/0.5 |
| 1 | CIBs functionality | 3 | F | 0/1 | 0/1.5 |
| 1 | EditTexts for pallet and location numbers creation | 2 | F | 0/0.5 | 0/0.5 |
| 1 | Keyboard behaviour | 3 | F | 0/0.3 | 0/0.2 |
| 1 | Keyboard actions | 5 | W | 0/0.5 |  |
| 1 | ListView creation | 2 | F | 0/0.5 | 0/0.3 |
| 1 | ListView functionality | 2 | F | 0/0.5 | 0/0.5 |
|  | ListView layout border | 4 | N | 0/0.3 |  |
|  | ListView long-click action | 3 | N | 0/1 |  |
| 1 | Buttons creation | 2 | F | 0/0.3 | 0/0.3 |
| 1 | Buttons functionality | 2 | F | 0/0.5 | 0/0.5 |
|  | TransferActivity | 1 | N | --- | --- |
|  | InventoryActivity | 1 | N | --- | --- |
|  | SettingsActivity | 1 | N | --- | --- |

# VII. Backlog

Sprint #1:

Setup Heroku-Parse server to be used as the backend database. Start working on the login screen, main menu and the two main functionalities - Inbound and Outbound. UI aesthetics are taken into consideration due to their simplicity and also to avoid coming back and re-doing all of them after the functionality is implemented. Test-driven development is based on bug avoidance and efficient use of database access.

Specifics:

* MainActivity is used as the login screen
* HomeActivity is the main menu of home screen
* InOutActivity is used both as the inbound and outbound screen

Sprint #2:

Work on the creation and functionality of the TransferActivity, and merely start working on both the Settings and Inventory activities. Polish the activities created on Sprint #1 in order to finish them for the time being based on the ideas gathered between sprints.

Specifics:

* TransferActivity is used to transfer products between locations
* SettingsActivity accesses the Settings dashboard which will lead to further actions mostly for administrators
* InventoryActivity acceses the Inventory dashboard which will lead to further actions within the virtual storage

# VIII. Change Log

Dec. 31, 2016 - Workbook is created and updated according to current development progress. Missing the Data Objects and Scrum sections!

Jan. 9, 2017 - Data Objects and Scrum are updated according to the development process after the first sprint is already done. Also update on what sprint #2 will be about.

# VIII. Glossary

User - Any person that is currently using the application.

Valid user - User with existing account.

Worker - Second tier valid user without full privileges. Valid actions include inbound, outbound, transfers and inventory consulting.

Administrator - First tier valid user with full privileges. All actions can be done including but not limited to inbound, outbound, transfers, inventory consulting, and the managing of both users and storage locations.

UI - User Interface

Activities - One ‘screen’ on the application

TextView - Shows text on the application

EditText - Text field for the input of data (e.g. text, integers, etc.)

ActionBar - Bar at the top of the application that can include actions (e.g. back button, settings button, etc.)

Button - A button on the application

ImageButton - A button with an image attached to it

ListView - A panel on the application that shows a list