Vending Machine

SOFT2412: Agile Software Development Practices

Sprint 1 Technical Report

Contents

1 Executive Summary

Sprint 1 of our project development revolved around the production of the most fundamental aspects of our Vending Machine application: the JavaFX GUI, a basic user hierarchy, and a persistent database storage solution for the software.

2 Scrum

2.1 The Team

The Scrum team is formed of three key roles: the developers, the product owner, and the scrum master. Each of these roles are committed to producing some pieces of work (each known as a *Scrum artefact*) in order to further the team towards the sprint goal. Each role is also accompanied by unique responsibilities which are briefly described below:

- **Developers**
  Developers are dedicated to completing tasks and implementing user stories to further the team towards an *Increment* every sprint. However, developers are also actively involved in the sprint planning and quality assurance of the project, and are expected to keep the Sprint backlog up-to-date as they complete tasks.
- **Product owner**
  The product owner represents the multitude of stakeholders involved in a software development project and attempts to focus the scrum team to produce a body of work which is the most in-line with what the project requirements are. Communicating the product goal to the rest of the team and, by extension, maintaining and prioritising the product backlog are tasks assigned to this role.
- **Scrum master**
  The scrum master is aptly named; they are the master of ensuring the scrum method is implemented correctly in the team. They must ensure the team understands and adheres to Scrum principles by facilitating stand-up meetings, sprint planning and discussion, goal setting, and ultimately aiming to produce a high quality Increment at the end of each sprint.

The team went through these roles and responsibilities in Sprint 0, as can be seen in the Confluence documentation excerpt below. Furthermore, the team was able to invite a real product owner who works at Westpac for a brief meeting where the team heard first-hand how each role operates in a business setting. See Box for the meeting notes from this meeting.

Excerpt of meeting minutes 09/10/22

11:15 - Professional Scrum Master Anil Dashmohapatra joined the meeting.

- The difference between the roles of Scrum Master and Team Leader was discussed.
- Anil highlighted that those with the engineers could have smaller inherent roles, such as ‘tester’ or ‘developer’.
- The project manager must have command and control and is the driver of the project. They must propel the plan and instruct others in a one-dimensional manner

Team Roles

📋 Overview

Identify and discuss team responsibilities by following the instructions for the Roles and Responsibilities Play.

Team	Agility Boyz
Team members	@Sulav Malla @Antriksh Dhand @Udit Samant @Ankit Kapoor @Nemo Gage
Date	08/10/2022
Team mission	To Deliver a Vending Machine App Agilely

📘 Roles and responsibilities

Roles	Responsibilities (SHOULD DO)	Responsibilities (SHOULDN'T DO)	Comments

Product Owner
@Sulav Malla
	• Has a very clear idea of entire product and requirements
	• Have a vision for the product placing them into user stories.
	• Prioritise user stories.
	• Maintains Backlog and issue priorities.
	• Communicate with scrum master at the end/beginning of each sprint.
		•	• Setting the high-level vision of the product.
			• Document this vision so it's extremely clear to the entire team.
			  ◦ End-to-end product flow
			• Sprint goals: "These features are what I want to see at the end of each sprint."
			  ◦ Each Jira card should **very clearly** refer to one of these objectives
			  ◦ There also should be an "acceptance criteria" for each of these objectives
			  ◦ The product owner checks these.

Scrum Master @Udit Samant	• Review and planning of the sprint • In charge of facilitating daily standup. • Motiviate everyone when they are feeling shit. Source of daily motivation.	• Manage developers • Does not manage people's sprints (team decides) •	• Replace "Project Manager" – command & control. ◦ Very one-directional/almost dictator • A "facilitator". He will not do everything himself, but he will organise meetings to make sure the work gets done/the decision gets taken. • The main people in the decision making are still the entire team. • Scrum Master is also a developer!
Engineers			• Everyone apart from the scrum master and product owner are equal. No segregation.

Unassigned responsibilities
Communication • Communicate early on very on when facing an issue with teamwork.
Dos • Be consistent. ***At least*** have a look at the code every day. Don't Dos • Do not keep things to yourself regarding the project • Do not say that you'll do it… ***Explain*** what you're going to do. No "I got this". • Do not assume everyone understands your idea.

Figure 1: Confluence excerpt: responsibilities of each role

in order to achieve success.

- The scrum master must not decide what is to be done. They must just ensure that decisions are happening and are being achieved, however decisions are up to the rest of the team. The scrum master is also a developer, and may take on other roles if necessary.
- Anil outlined the concept of adaptive planning. He stated that planning does not have to be absolute or complete before progressing to the next stage of the development cycle. In adaptive planning, it must be made sure that prior to the first sprint there is a high level of clarity between team members on the approach being taken to the sprint.
- The role of documentation in an agile development environment was discussed.
- Anil stated not to follow the standard documentation process of other approaches such as the waterfall approach. The correct approach to documentation is to focus on flexibility.
- Anil commended our use of Jira and Confluence. He recommended to make use of the features of both applications and to only document using bullet points, as there is no need for extensive documentation. It should remain minimal.

Table 1: Distribution of roles in the scrum team

Member	**Role**	**SID**
Antriksh Dhand	Developer	510415022
Nemo Gage	Developer	500496851
Udit Samant	Scrum Master	500700976
Sulav Malla	Product Owner	500495980
Ankit Kapoor	Developer	510470180

2.2 Sprint Goals & Overview

Sprint 1 is actually trailing off the back of "Sprint 0" – a setup sprint conducted between Weeks 9 and 10. Sprint 0 was essential in planning and prioritising our development time; during this sprint it was decided the goals for Sprint 1 would be to develop a) a basic JavaFX application with some beginnings of a GUI, b) a perpetual database with user login capabilities, and c) the user hierarchy described in the product requirements.

2.3 Product and Sprint Task Boards

Quality of the way the team created, managed, and maintained the Product and Sprint Backlogs in accordance with Scrum Practices. Following the first meeting on Thursday, 6th of October, the team concluded to proceed with Jira Software to manage our Product and Sprint Task Boards, our backlog, and our documentation. Before discussing how Scrum Practices were implemented on Jira, this portion of the report will first address why the Jira was the chosen tool. There are a large number of Agile Management tools. However, Jira is the only one that implements Scrum Backlogs and Task Boards in a manner that easy to learn and apply. Using a traditional Kanban Board, would mean that planning sprints, and working on the current sprint would be difficult to organise. It becomes difficult to different which user stories, and tasks that are being worked on in the current sprint, and which one are being planned for the next sprint. Additionally, while Jira offers an astronomical number of features that would take an incredibly large amount of time to fully apply, one does not need to use all them to apply Agile Development to a project.

Even within Jira there are various types of boards and to use to plan Product and Sprint boards. The board that was collectively decided upon for this project, was the Scrum Board. The reasoning is that the Scrum Boards, still offers a Kanban Board, however the Kanban Board is now focused onto the current sprint. That way maintaining a backlog becomes easier and more organised.

The first functionality that was applied using Jira was the use of Epics. The team first constructed Epics, their expected ending, and their descriptions. Epics were implemented as larger, broader ideas that needed to completed over multiple sprints.

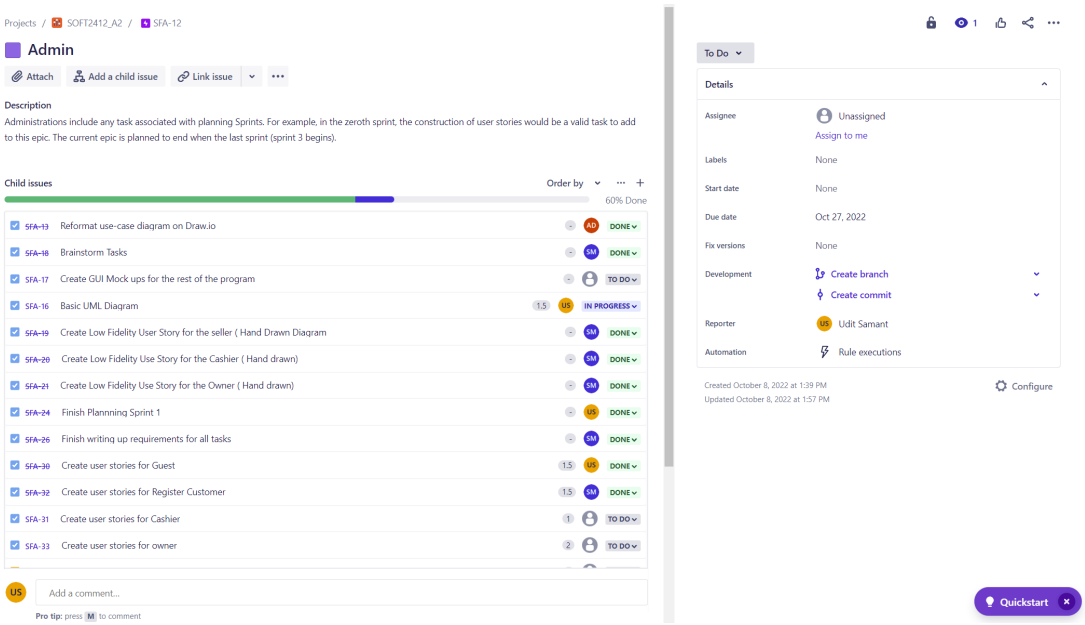


Figure 2: Screenshot of our “Admin” Epic

Then, the Scrum Board tasks were divided into 4 main categories: task, story, bug, idea. Tasks are any task that was not on the implementation level. For example the creation of a use-case diagram would be a task. A story was implemented as user story. Each user story was given acceptance criteria and objectives which made clear what should be tested. Bugs were a custom tasks defined to be any issue that could not be solved with one individual in one sitting and needed documentation. Lastly, ideas are features that is to be presented to the client as extra functionality that would improve the value and experience of the app. The team presented one of these features ("As a owner I should be able to elevate privileges of a user, so that I do not have to add a new seller/cashier when an account already exists.") in the first client meeting.

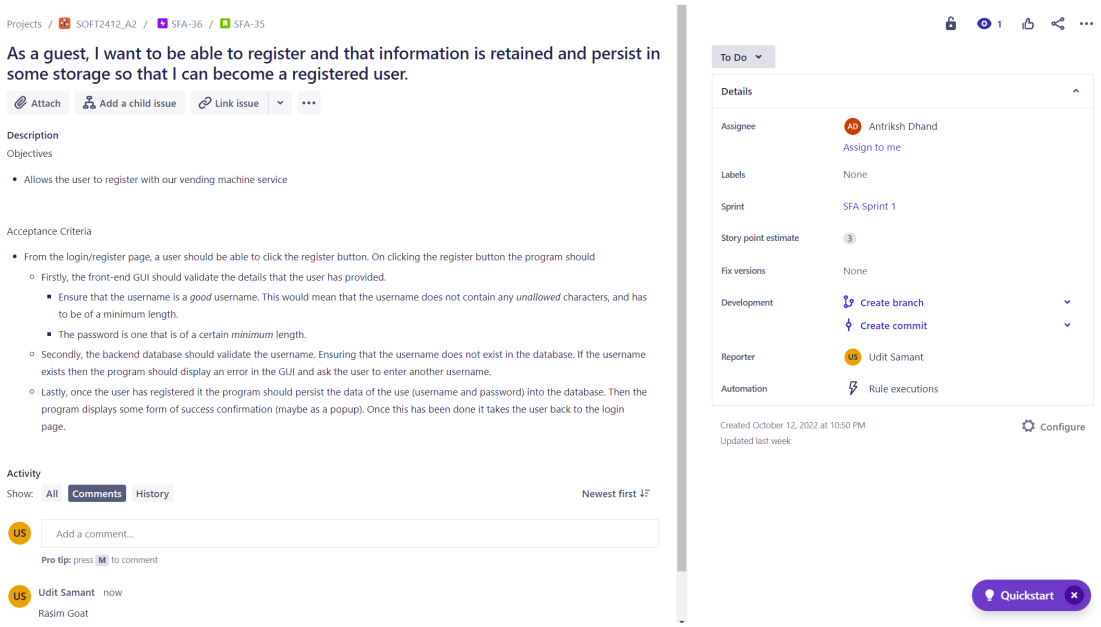


Figure 3: Example of a user story

Now that the structure of the Jira tasks have been addressed, this paragraph will address the usage of the two boards available to the team. The two boards were the Scrum Board and the Sprint Board as can be seen below. The Backlog/Scrum Board was used to plan tasks identified for the next sprint and the sprints following. Here user stories and the user story acceptance criteria were created. Then once the sprint begun the team utilised the Kanban board. As seen in the third Screen shot below the the Kanban board follows a very typical structure. The additional column check requirements and its two sub categories refer to when a task needs to either get tested (using JUnit) or confirm that it meets acceptance criteria directly if it is code that cannot be tested or tasks not involving coding.

2.3.1 Documentation

For documentation the team linked the Jira page to a confluence environment and used confluence to note any decisions and meeting notes that we had. This allowed for easy access to all past meetings and avoided any discrepancies or confusion in the future, for any time a team member was unsure of what had been discussed, they could easily refer to the corresponding meeting's notes. Meeting notes were kept brief and to the point to fulfil the purpose of an outline of the topics discussed in each meeting.

2.4 Artifacts

2.4.1 Standups

Through out the first sprint, the team held daily standups, where the team would have short 15-20 minute meeting where each member of the team would share their progress and what they are planning to do next. The times of each stand up was decided the day before on the day of and was very flexible, as finding an agreed set time for a stand up was too difficult to manage as a 5 man team. Each stand up was recorded, with all event that occurred within the being scribed by a member of the team whom would alternate each stand up. All the notes from the scribed notes form these meetings have been added to the appendix.

2.4.2 Sprint Review

On the last day of the sprint, a sprint review as conducted. It was lead by the Scrum Master and feedback on how the team performed over the course of the sprint, improvements were suggested and each member got to self-reflect on how they thought they went and reflect on the feedback they were given. The events of sprint review were scribed and can be found in the appendix,included in the meeting notes.

3 Agile Development Practices

3.1 Git

3.1.1 Webhooks

Git webhooks were attached to several platforms to ensure that all of the software development tools were integrated with each other.

The webhooks used were as follows:

- Jenkins
- Discord
- Jira

The Jenkins webhook was used to monitor build success and iteratively document each push to the repository. Jenkins was further integrated with Discord to notify the team whenever versions were updated and to verify that builds were successful on each push of the program.

The Discord webhook was used to alert team members whenever team members had used the merge, commit and push commands to connect to the repository. This particular webhook was attached to a Discord channel that would message the team server any time the aforementioned functions were used to remove the need for extensive messaging upon changes to the application and to alert members when a pull to their local machine was required before any amendments were made.

The Jira webhook primarily existed for the team's convenience. It allowed for specific commits to be attached to issues to streamline the completion process of all tasks on the project board.

3.1.2 Commit Messages

Under Antriksh's strict command, team members followed a set commit message and description structure for all commits sent to the server. The format was as follows:

```
<present_tense_verb> <task_completed>
```

This format aligned with that of Git's automated merge and push messages to maintain consistency throughout all messages in the repository.

3.1.3 Pushes to Main

Team members collectively decided to primarily work on the main branch to conform to the agile software development approach, as recommended by the group's tutor, Rasim. This ensures that changes were consistently sent to the main branch and quickly accessible by all team members

upon completion with minimal merge conflicts. If features were in the process of being implemented and the build was not successful upon a particular commit (as detailed by the Jenkins webhook), team members were encouraged not to push but rather continue committing upon completion of minor changes, ensuring that the main application could still build correctly as often as possible. Merge issues were not an error as team members would work on different Java files, as they were purposely assigned tasks that were to be completed on separate files in order to mitigate issues when combining functions and features.

3.2 CI/CD Tools and Practices

GitHub, Gradle, JUnit, Jenkins

For this project and this sprint in particular the team created a pipeline using Gradle, Git and Github, Jenkins, and Discord.

- Gradle
  Gradle was used to develop and build the the program. It was customised to suit the needs of the team, adding the correct dependencies need to build and implement the project.
- Git and Github Git was used for collaboration, each team member was has access to a remote repository where they could pull, make, and push change so each team member always had the latest version to the project.
- Jenkins
  Jenkins was used for the purpose of final unit and end-to-end testing and deployment of the application. We set up jenkins to do the gradle tasks below, and then test all the code below.

- Discord and Communication Discord was used as the main form communication, all meeting and stand up were held on discord and other forms of communication were also done there.

4 Application Development

4.1 Group Collaboration

The group collaborated to complete the Vending Machine application using various tools, including GitHub, Jira, Gradle, Jenkins and Discord. These tools made it easier to collaborate, with automated testing, building and version history. The individual contribution was good across all members, Udit worked on the Database and Seller Portal, Nemo completed the Main Default Page, Suluv completed the Owner Portal, Ankit finished the Cashier Portal and Antriksh make the login functional.

The group communication was excellent with daily standups; see minute notes for details.

4.2 Application Design

The following is an overview of the Vending Machine Application Front-End design.

A Meeting Minutes

This section contains all meeting minutes taken over Sprint 0 and Sprint 1.

Meeting Notes

06/10/2022

The meeting started at 04:30 PM at J12.

Written By - Udit

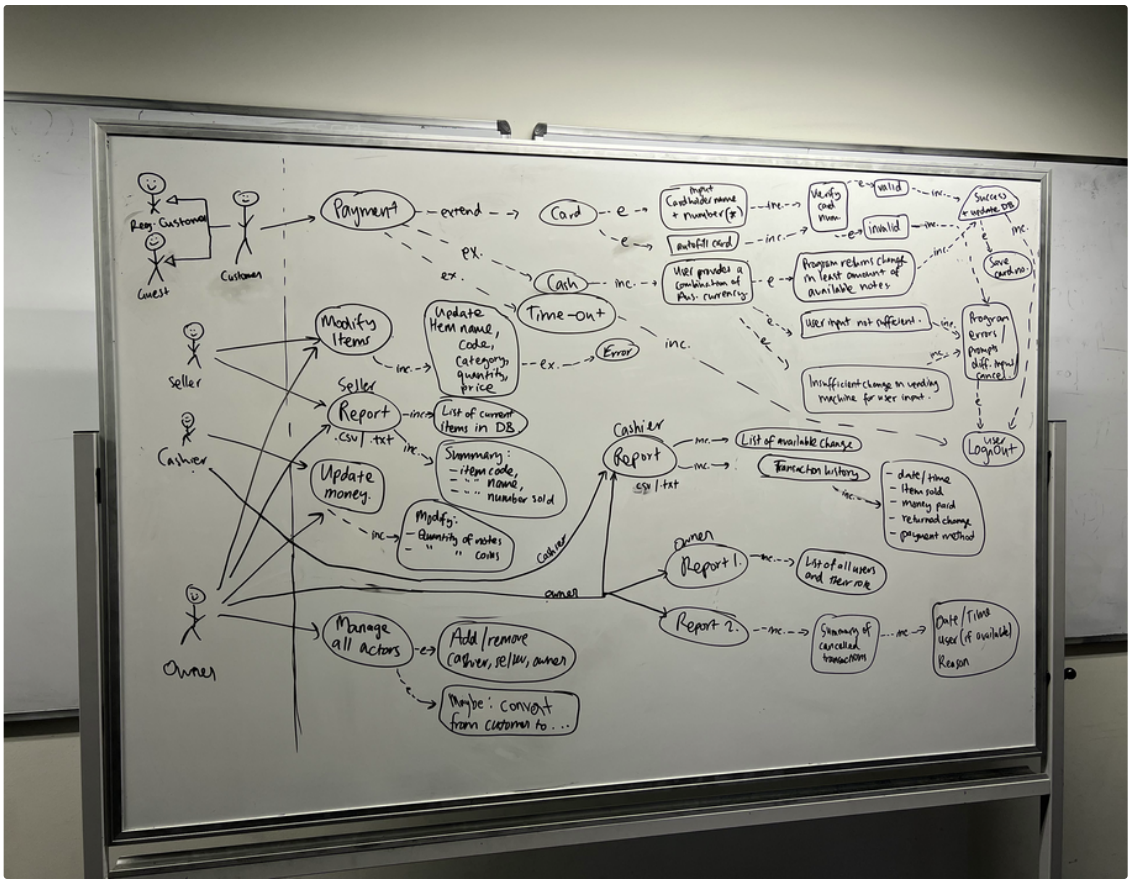
Attendees: Nemo, Sulav, Antriksh, Udit

Meeting notes:

- Decided roles
  - Client - Rasim (Tutor)
  - Product Owner - Nemo
  - Scrum Master - Udit

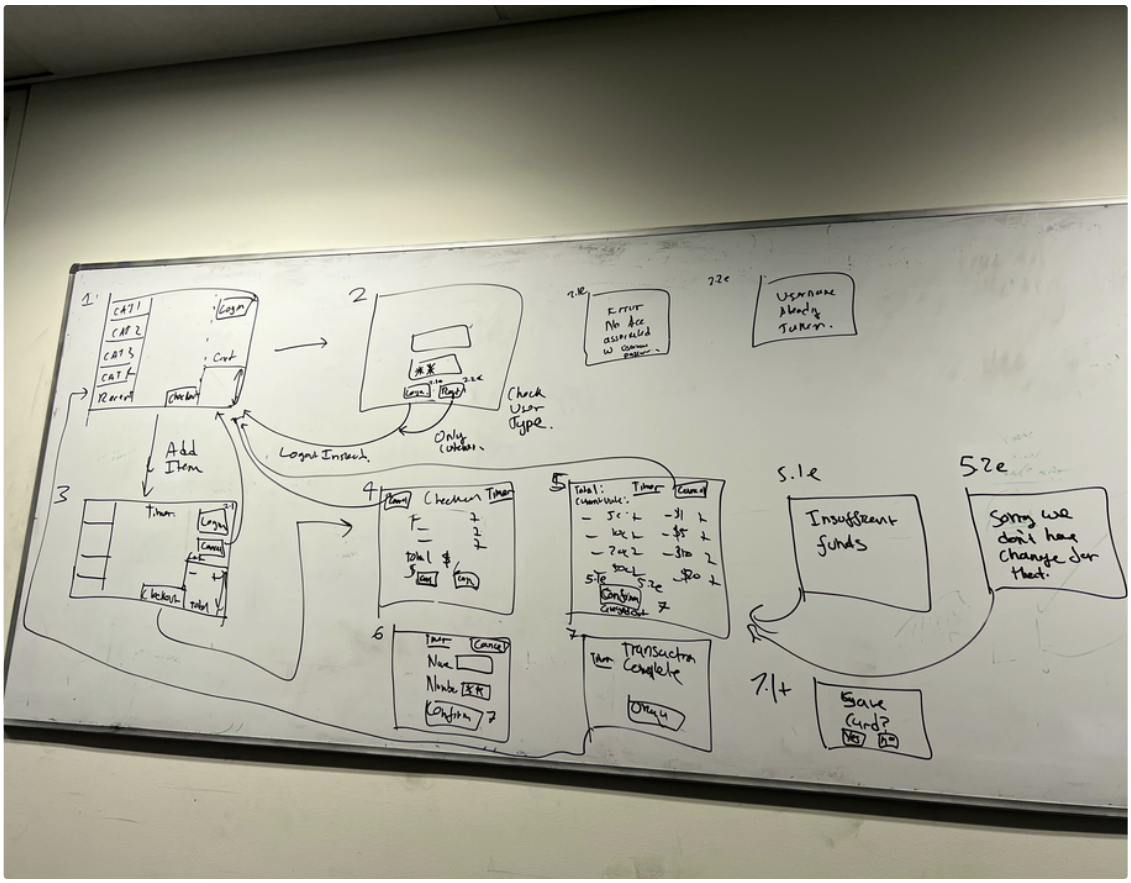
- Reflect on previous assignment and what we can do to improve in this one.
  - Clear communication
  - Clearly defined roles
  - As many short but quick meetings as possible

- Created a use-case diagram



Use-Case Diagram

- Created basic low fidelity mockups of the user interface (for registered customers and guests)



Lo-Fi Mockup - Reg Customer/Guest Customer

07/10/2022

Attendees: Nemo, Sulav, Antriksh, Udit

Written By - Antriksh

Meeting notes:

- This was the team's first stand-up meeting.
- No coding is allowed until Thursday, hence it was decided that we will use the time until next week to get all the project requirements cleared.
  - In real software projects no coding is done until everyone has a clear idea of what the app with look like and what the requirements are.
- It was decided that we will use Jira to manage our scrum.
  - Briefly discussed what the purpose of each role in a scrum is e.g. scrum master is not a leader, is more like a logistical person which ensures things happen. Scrum master ensures sprint review, ensures sprints occur and stand up meetings occur.

@Ankit Kapoor also had requested to join the team at this point, and we were discussing the possibility of such. We decided to revisit the topic the next day with fresh minds.

08/10/2022

The Meeting started 12.30 PM.

Written By - Sulav

Attendees : Antriksh, Sulav, Udit, Nemo, Ankit

Meeting Notes:

> Major Decision : Ankit was allowed to join the team if he wants.

The Roles for each person were assigned, and what each role involves were defined as a group and the responsibilities were defined.

12.57 pm, Ankit joined the call. He was added to the relevant repositories and software the team is going to be using through out the project.

Since the sprints haven't started the stand up meeting are going to take longer than expected.

The Do's and Don'ts for each role and the team holistically were set up. The Scrum Master debriefed the whole team on Jira and how the project management is going to run through out the sprints.

Tag

- Minor user story is a new minor patch * . * . #
- Major/ epic changes is a new major patch * . # . *
- Major Epics is a new #. * . *

0th Sprint has begun and will go from today to the next Thursday.

The fist epic was created, "Admin" and first few tasks are created.

Next Meeting will be held on 9/10/2022 at 11 am, and a few tasks were delegated to some of the member and should be completed before the next meeting. (Look on Jira for the tasks)

09/10/2022

The Meeting started 11.00 AM.

Written By - Ankit

Attendees : Antriksh, Sulav, Udit, Ankit

Meeting Notes:

Each attendee discussed the work they had completed and what they plan to do before the next meeting.

The agile manifesto was discussed.

It was decided that when creating an issue, the description should have a concrete description of the issue and what it means to streamline the resolution process.

11:15 - Professional Scrum Master Anil Dashmohapatra joined the meeting.

The difference between the roles of Scrum Master and Team Leader was discussed.

Anil highlighted that those with the engineers could have smaller inherent roles, such as 'tester' or 'developer'.

The project manager must have command and control and is the driver of the project. They must propel the plan and instruct others in a one-dimensional manner in order to achieve success.

The scrum master must not decide what is to be done. They must just ensure that decisions are happening and are being achieved, however decisions are up to the rest of the team. The scrum master is also a developer, and may take on other roles if necessary.

Anil outlined the concept of adaptive planning. He stated that planning does not have to be absolute or complete before progressing to the next stage of the development cycle. In adaptive planning, it must be made sure that prior to the first sprint there is a high level of clarity

between team members on the approach being taken to the sprint.

The role of documentation in an agile development environment was discussed.

Anil stated not to follow the standard documentation process of other approaches such as the waterfall approach. The correct approach to documentation is to focus on flexibility.

Anil commended our use of Jira and Confluence. He recommended to make use of the features of both applications and to only document using bullet points, as there is no need for extensive documentation. It should remain minimal.

Some other notes from the meeting with Anil:

(Agile) Adaptive planning:

- Waterfall — everything happened in a sequence. Analysis, design, AND THEN development…
  - Every stage has to be 100% done before moving on to the next
- Plan (requirements/mockups/etc) should be ~80% done before the first sprint.
- In the second sprint, planning should be less, e.g. ~60%
- Planning = “Clarity of what you’re going to do”
- Have the big picture of what the product will look like
- High-level plan of where you start and where you end should be essentially confirmed
  - If after 2 sprints you realise the high level design you have put in place isn’t going to work = problem
- Big picture planning should be done. Small things are less important.
- e.g. you have a UI page A and UI Page B. You know how these are going to interact = big picture.
  - But within UI page A, you have 2 features that were there that won’t be in the future. That small change is okay.
  - These small changes can be easily factored in.

Documentation:

- Waterfall — heavy documentation e.g. High level design document…etc.
- Agile — need to create documents which user can derive value out of.
  - Need to have a history. “Why did we make this change in Sprint 2?”
  - Jira cards are also good for documentation.
    - Description tab/acceptance criteria tab/testing tab. These are also documentation!
- Brief documentation is good for a small team.

10/10/2022

The Meeting started 10.00 PM.

Written By - Udit

Attendees : Antriksh, Sulav, Udit, Ankit

Meeting Notes:

- Quick Standup
  - Antriksh, Ankit, and Udit do not have progress updates from yesterday as they have had assignments. However, Antriksh has completed his assigned task.
  - Sulav has described what he has done as with LoFi Mockups and blueprints for user storeies.
    - Has gone over issues with our current user stories and product requirements.

- Has gone over his designs for the GUI Mockups for login, default page, seller portal, owner portal, and a few extra pages.

11/10/2022

The Meeting started 11:00 PM.

Written By - Nemo

Attendees : Antriksh, Sulav, Udit, Nemo, Ankit

Meeting Notes:

- Generally not much to contribute
  - No updates - everyone has done no work
  - Storyboard update - Ankit relayed that he had begun the HTML storyboard and was implementing the pages that the group had decided on in previous meetings

---

12/10/2022

The Meeting started 10:30 PM.

Written By - Udit

Attendees : Antriksh, Sulav, Udit, Nemo, Ankit

Meeting Notes:

- Standup
  - Ankit - HTML storyboard had been completed for the main pages of the GUI.

13/10/2022

The Meeting started 02:00 PM.

Written By - Ankit

Attendees: Ankit, Antriksh, Udit, Sulav, Nemo

Meeting Notes:

- The group finalised their notes for Sprint 0 and began Sprint 1.
  - The expected results at the end of Sprint 1 were discussed in order to prepare for the upcoming week's tasks.
  - The goals of the team were descried for the upcoming sprint.
- Udit and Antriksh presented their conflicting interpretations of the client's incremental needs, and a compromise was reached between their ideas.
- The features that would be implemented by the end of Sprint 1 were discussed.

14/10/2022

The Meeting started at 4:11 pm.

Written By - Sulav

Attendees: Udit, Ankit, Sulav, Nemo

Meeting Notes:

- The group went around and updated the group on what had been done between yesterday and today.
  - Udit went around assigning the tasks for each person.
  - How the login and register page looks have been posted as a mid fi user stories.
- Assignment 1 was shared with Ankit.
- Ankit was taught how the team wants everyone to be to structure their commit messages.

15/10/2022

The Meeting started 10:10 AM.

Written By - Antriksh

Attendees: Antriksh, Udit, Sulav

Meeting Notes:

- Antriksh will start working on the GUI for the login page.
- Udit finished all the story cards yesterday.
- Udit will finish setting up the SQLite database today and then start working on the GUI for the Seller's portal.
- Sulav has completed the Mid-Fi mockups for a lot of the screens including default page, log in, error etc. See this on the project pages.
- Sulav asked the rest of the team for their opinions on some design choices such as placement of buttons, linkage between screens etc.
- Udit and Sulav fixed up the acceptance criteria yesterday for a lot of the user stories.
  - Acceptance criteria should be much more concise than as they are now OR we should use "Scenarios" in our description of user stories.

16/10/2022

The Meeting started 9:00 PM.

Written By - Nemo

Attendees: Nemo, Udit, Sulav, Ankit

Meeting Notes:

- Udit had been working on the database and started writing code for it.
- Udit explained to Ankit how to query the database and how to write SQL in JAVA.
- Sulav had completed the acceptance criteria for most of the user stories.
- Ankit has started to code the cashier default page.
- Nemo has started to work on the main user default page. Udit explained to him how it can be low fi with even bullet points instead of categories.

17/10/2022

The Meeting started 11:00 PM.

Written By - Ankit

Attendees: Nemo, Udit, Sulav, Ankit, Antriksh

Meeting Notes:

- Ankit displayed his work in converting the program into a working JavaFX application.
- Ankit further demonstrated his development of a Cashier Portal console.
- Udit presented his updated database files and the amendments that he had made to the Seller Portal.
- Sulav explained how he had added functions to allow for the owner to operate the required functions.
- Antriksh outlined that he would complete the work that he was assigned before the next meeting.
- Nemo described his use of boxes and panes.
- The group educated Ankit on the developmental decisions and conventions that were decided upon regarding the project prior to him joining the group, such as commit messages, class extensions and programming styles.

19/10/2022

The Meeting started 11:00 PM.

Written By - Antriksh

Attendees: Nemo, Udit, Sulav, Ankit, Antriksh

Meeting Notes:

- Udit has completed the Seller portal
- Udit needs to communicate with Nemo to work on how we're transferring login information throughout the application
- Udit will start on the Sprint Report tonight
- Sulav mentions that actually *everyone* should be working with Nemo
- Sulav will be finishing his scenes for the Owner portal
- Ankit has replaced a lot of magic values in the program with constants
- Ankit has fixed up some x and y values to make the application look nicer (more like a page than an unordered mess)
- Ankit has essentially finished his tasks for this sprint (buttons on Cashier portal). He's looking forward to tomorrow's meeting with the client and starting the second sprint.
- Clarification by Udit: anything on the default page is assigned to Nemo. Do not work on other people's tasks!
- Nemo has been working on the same thing as Udit — making the scenes transition into each other whilst maintaining login information
- Nemo will be working on production and report tomorrow
- Antriksh is approximately 40% complete with the login page.
- The team had a brief discussion on what the sprint report should look like.
- The team had a short discussion on the implementation of the categories in the database.
- Udit: "By the end of next sprint, hopefully everybody knows how the database works."
- Stand up complete in a record time of 16 minutes.

20/10/2022: Sprint 1 Review

The meeting started 4:10 PM.

Written By - Antriksh

Attendees: Nemo, Udit, Sulav, Ankit, Antriksh

Meeting Notes:

- Stand-up meeting review
  - Meetings are too long! They usually last for one hour and take too much time out of our lives when we should be coding. Stand-ups should last 15 minutes max.

    - Try not to interrupt each other. The meetings are not supposed to be technical at all, and should be more like a casual catch-up.
    - Don't ask specific questions to other people regarding the project *in the stand-up.* This can all be done in a separate meeting outside the stand-up meeting time.
    - A discussion took place regarding the time of the stand-up meetings. But unfortunately due to the size of our group, it is difficult to standardise a time for these meetings.
        - All group members should *try* and join the stand-up meeting daily.
        - The next day's stand-up time will continue to be decided at the end of today's meeting.
    - Should we have a stand-up everyday?
        - The other group (which was very successful) only met 4 times in the week.
        - The argument is that the stand-up meeting is only meant to be 15 minutes, casual, and allows everyone to sync their mindset with each other.
        - *Other* meetings can be set up by individuals to code together or work on implementing features together.
- How have we used the tools we have attempted to use? (Jira, Jenkins, Git, VSC vs IntelliJ)
    - Jira
        - Team members express that they used the Jira backlog more than the Jira board.
        - Udit mentions that the backlog is meant to be used more for *planning* while the board is meant to be used day-to-day.
        - Ankit mentions the board isn't as useful as the backlog.
        - Nemo says he accidentally did some tasks that weren't assigned to him on the backlog.
        - Sulav mentions that at the start of each stand-up meeting someone should share screen and show the Jira board, so that everyone at least sees the board daily.
        - When you mark an issue as completed, ensure you place the commit number in that issue for tracking purposes.
    - Git
        - ```
          git add
          git commit
          git pull
          ```
        - These three functions are to be used more than `git push` to ensure you're staying up to date with the rest of the codebase.
        - Ensure you commit more frequently and push when you've successfully implemented a feature with a nice(!) commit message.
    - Jenkins
        - Please check the Jenkins build reports *at least every time you push something to Git.*
- Bug tracking with Jira
    - At the start of every sprint you should attempt to note down all the bugs present in the current product.
    - Major bugs (bugs that you need to discuss with someone else) should be documented on Jira using the "Bug" issue option.
- Sprint Review
          ```

B User stories

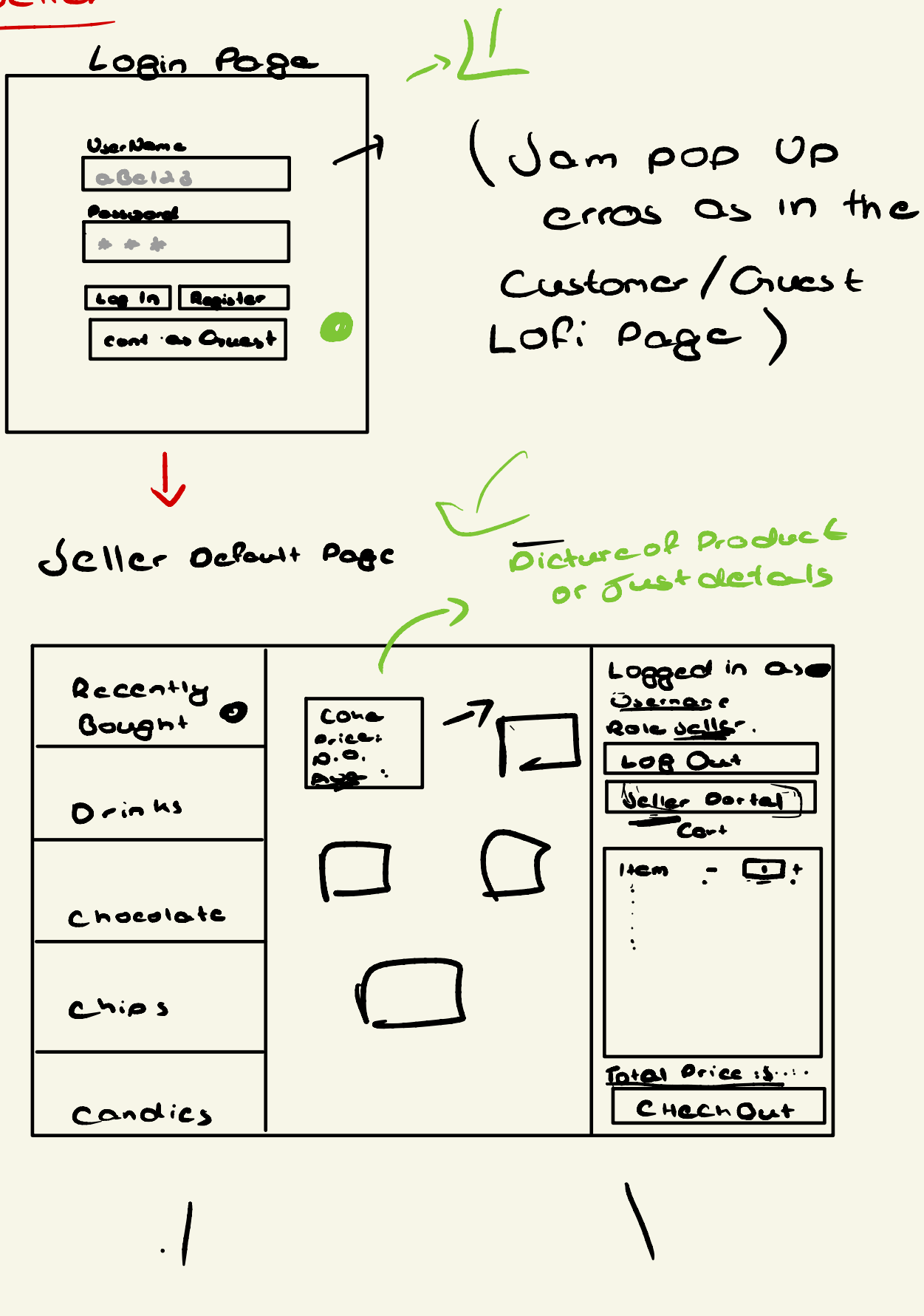
B.1 Lo-fi user stories

These lo-fi user stories were generated very early on in Sprint 0. These were then superseeded by the hi-fi user stories which can be found in Appendix B.2.

Product Owner Notes

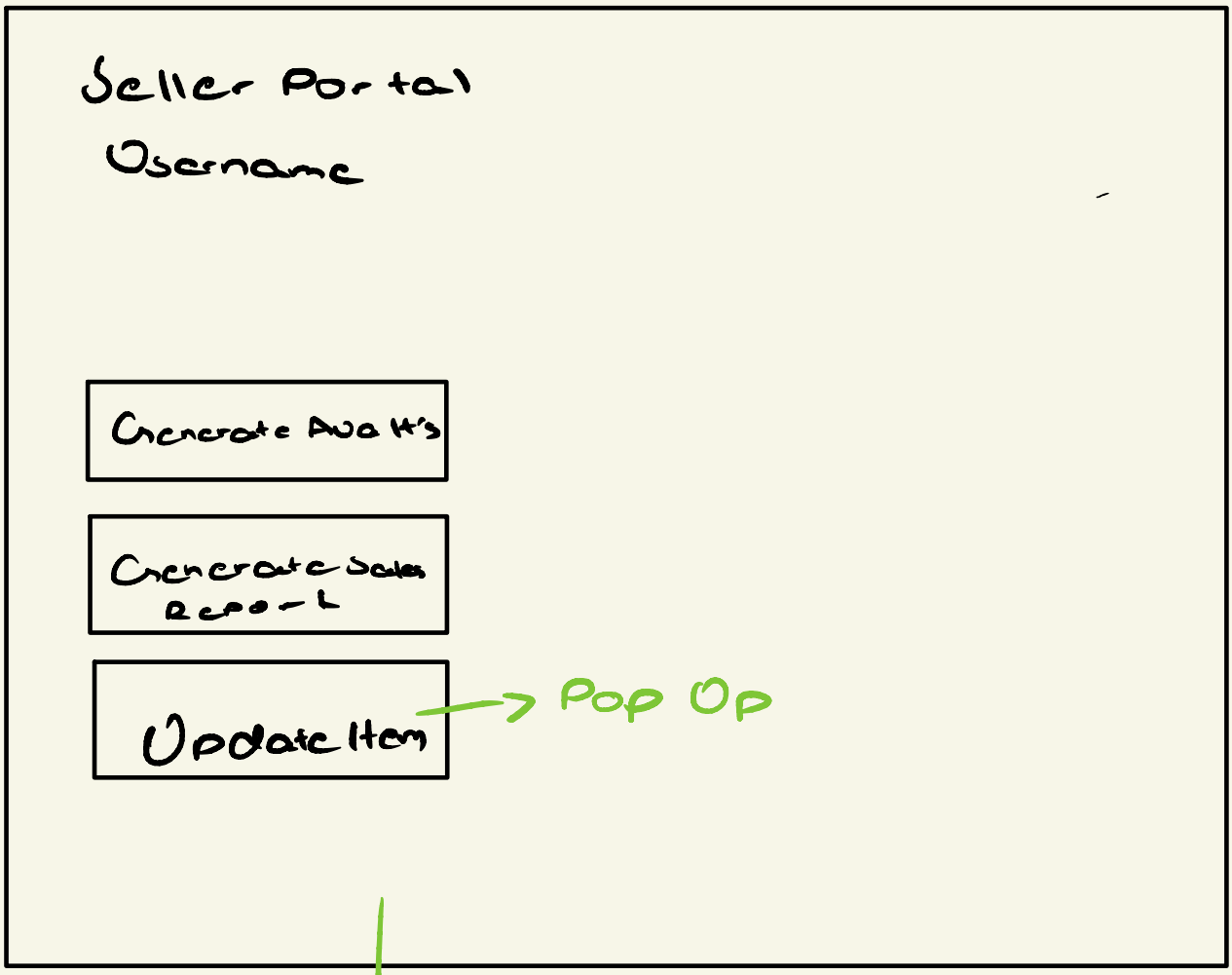
- Idle activity of 2 min is cancelled
  - Transaction → Default page (empty cart)
  - → Cancel can only occur during transaction. → Default Page
  - ↳ Both lead to empty cart
- → Automatic login (must log out seller → log to change)
- → last 6 products each time

Seller

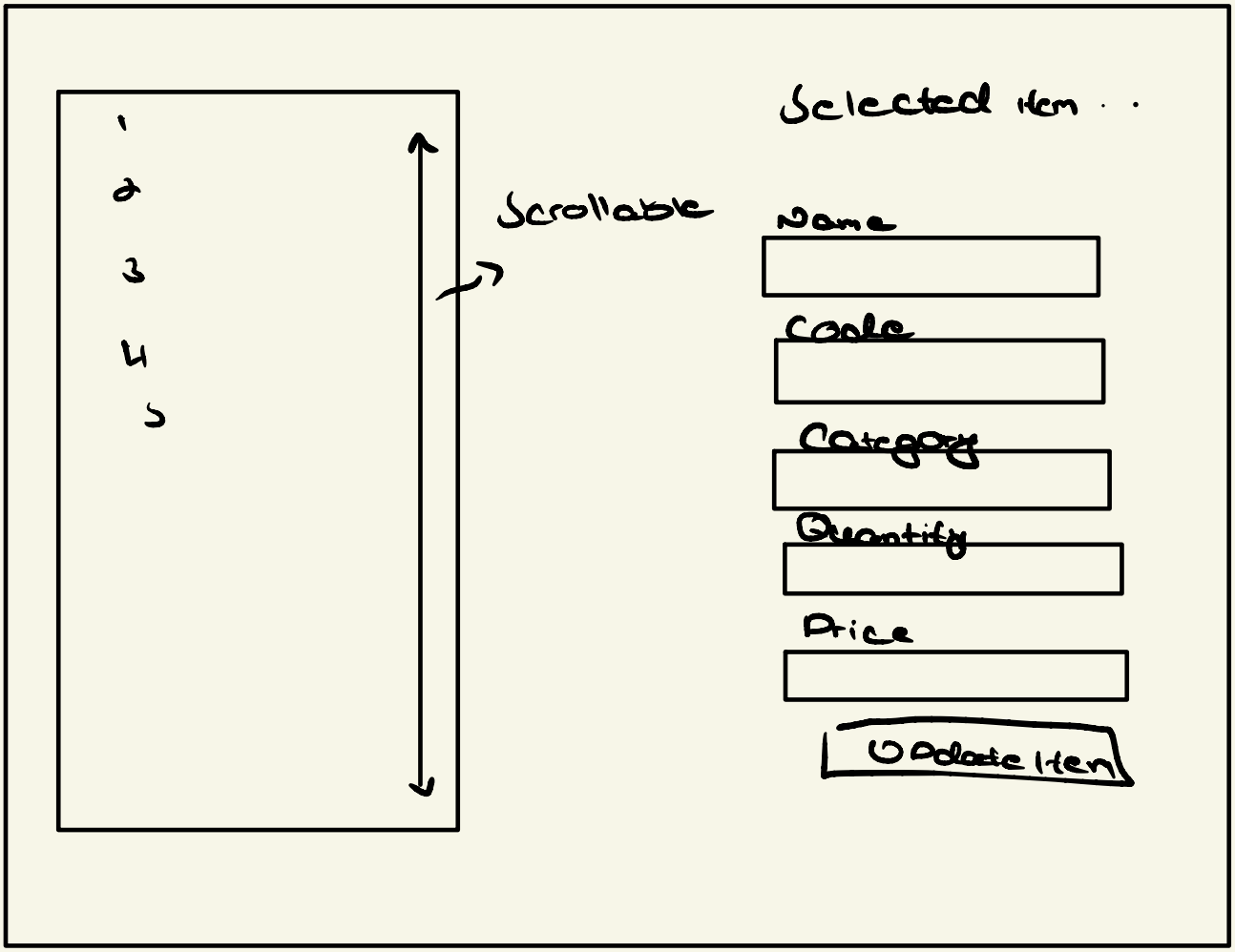

Login Page
UserName
abc123
Password
Log In
Register
cont as Guest
(Jam pop up erros as in the Customer/Guest Lofi Page)
Seller Default Page
Picture of Product or just details
Recently Bought
Drinks
Chocolate
Chips
Candies
Coke
price:
p.o.
Logged in as
Username
Role seller.
LOG Out
Seller Portal
Cart
item
Total Price is...
CHECKOUT

Transaction as Normal for Guest ..

Seller Portal



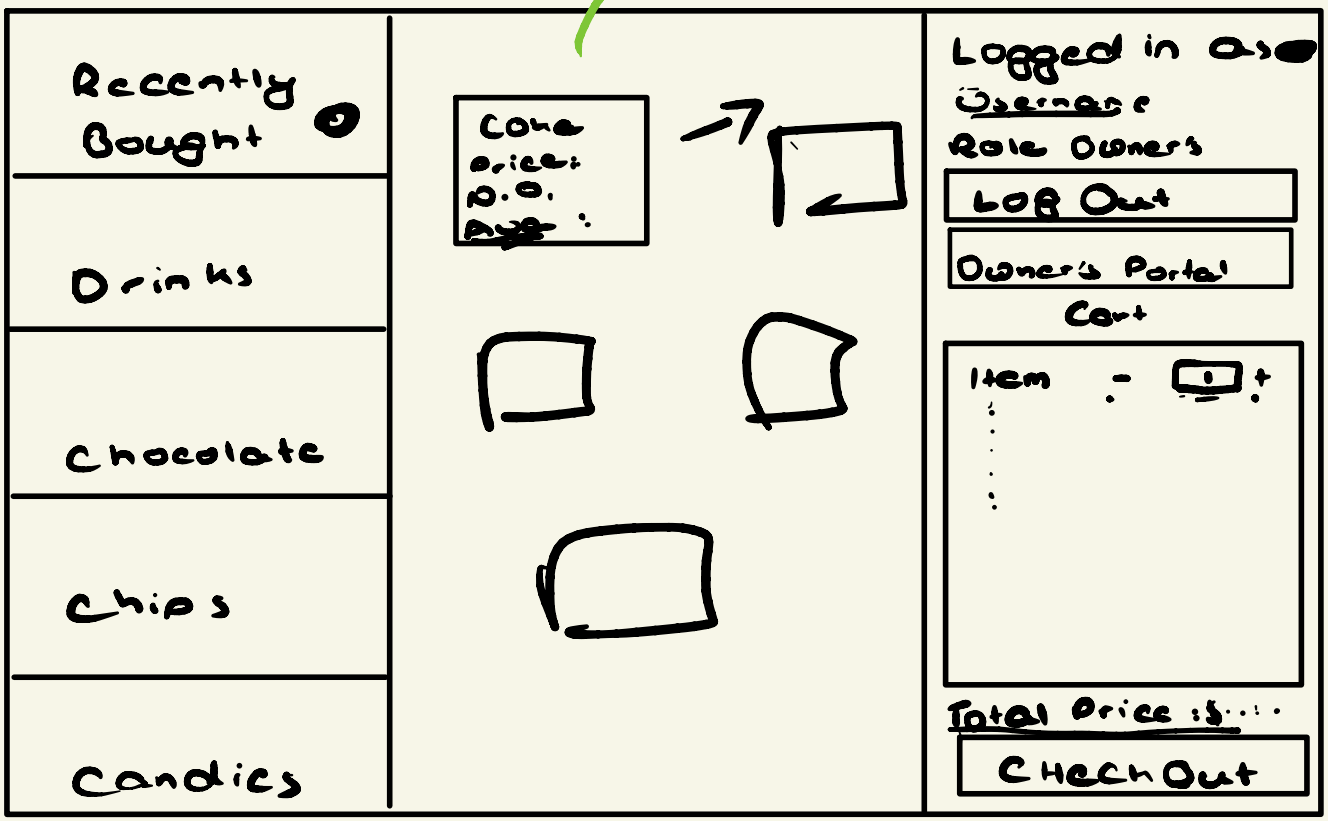
Pop Up


1
2
3
4
5
Scrollable
Selected Item . .
Name
Code
Category
Quantity
Price
Update Item

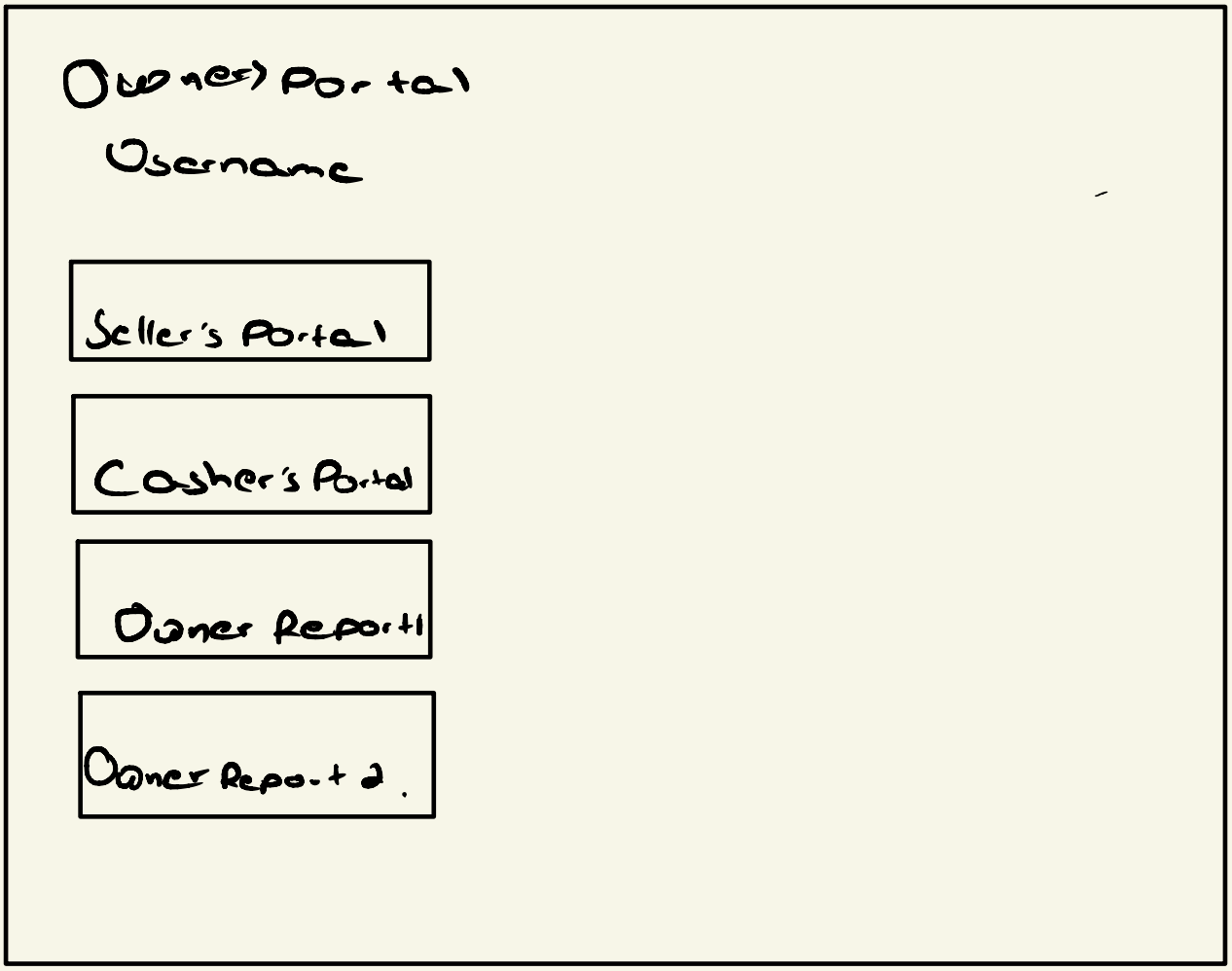
Owner's Portal

Seller Default Page

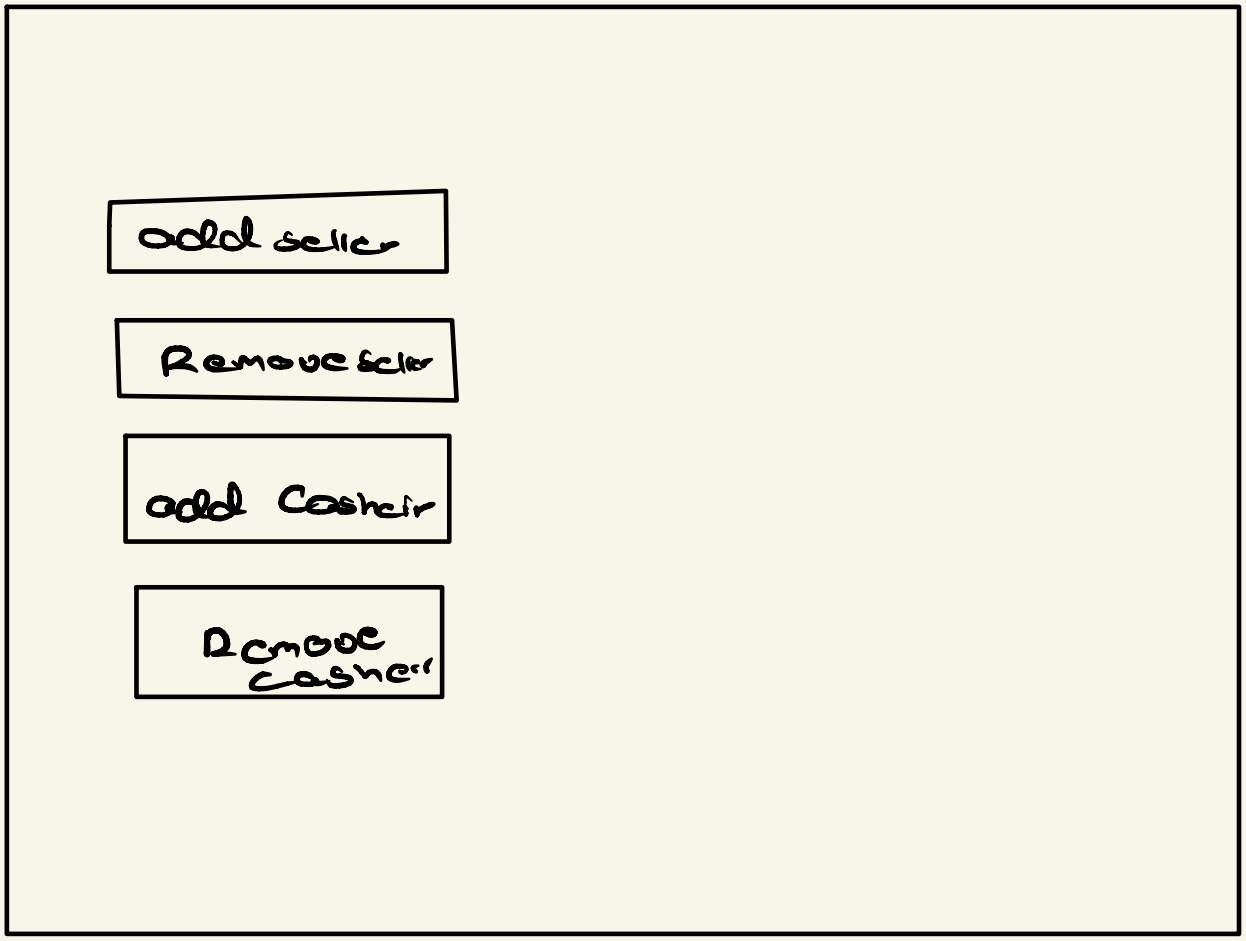
Picture of Product or just details



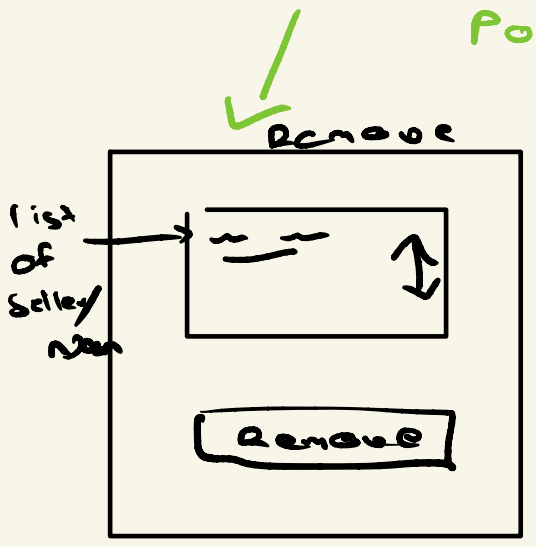
Owners Portal

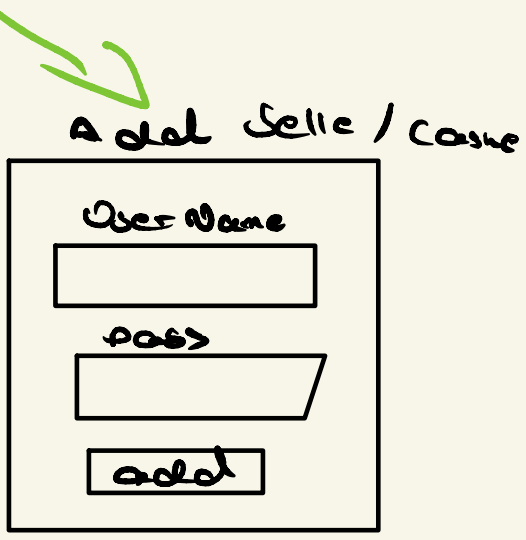


Owners Portal



Pop Up

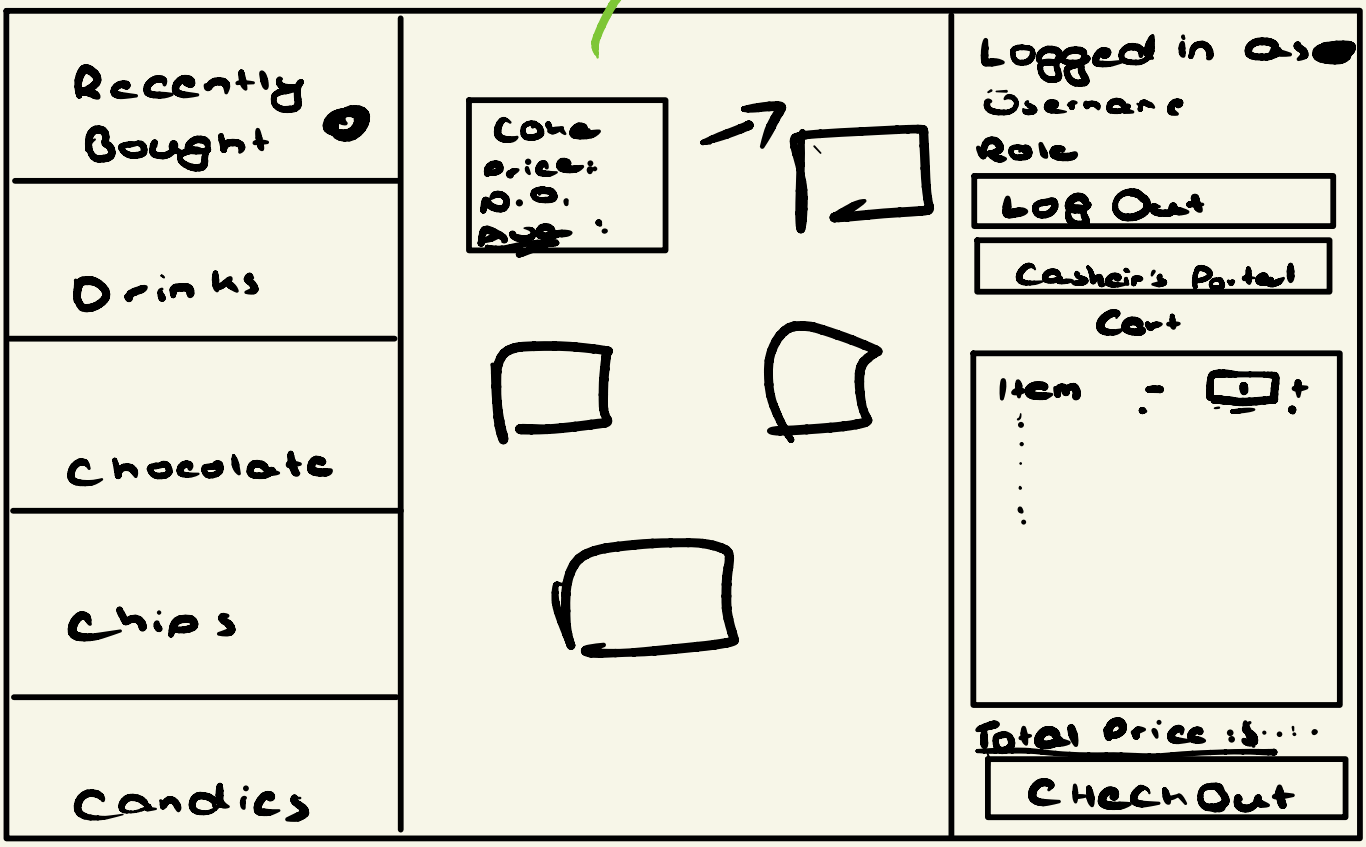


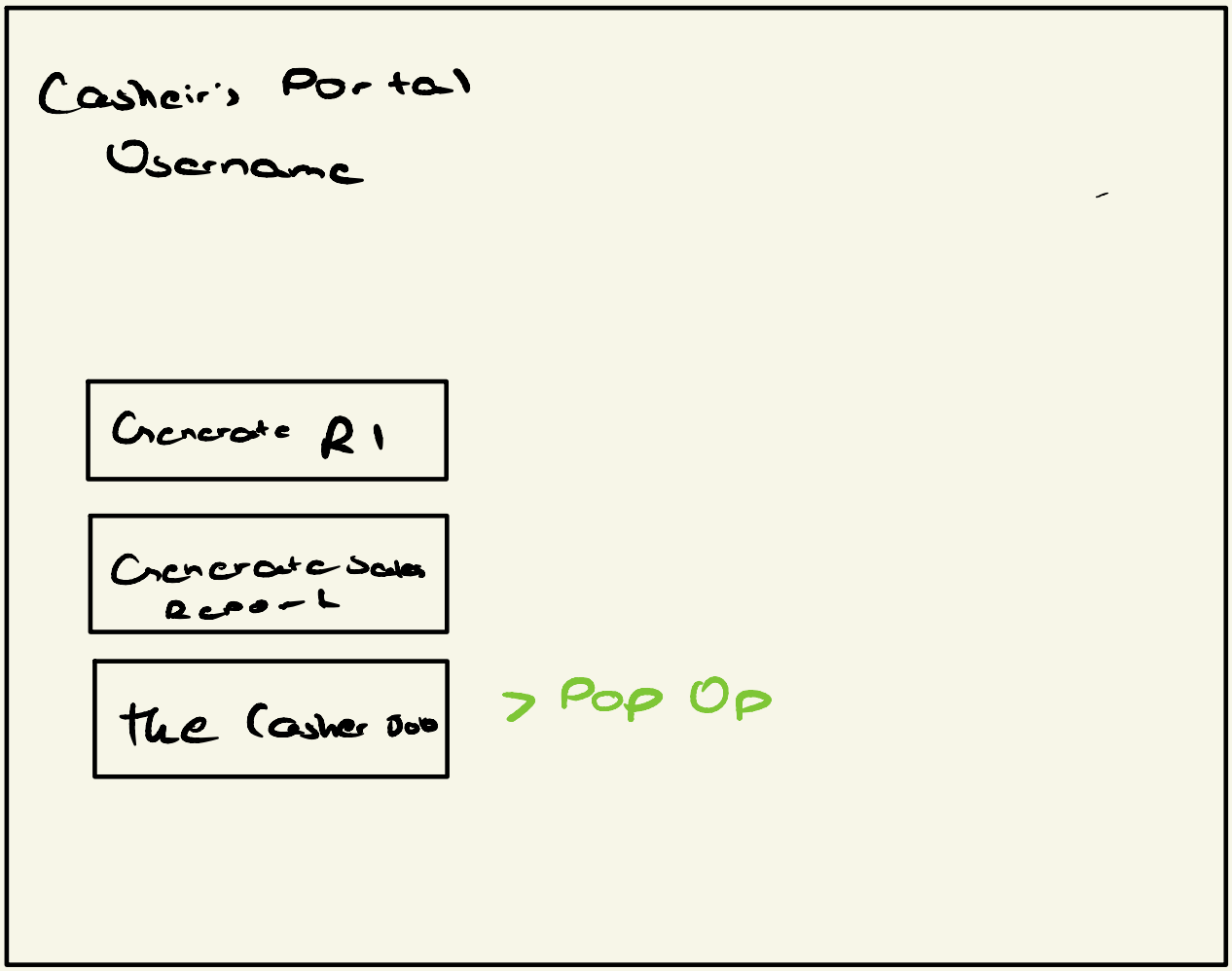


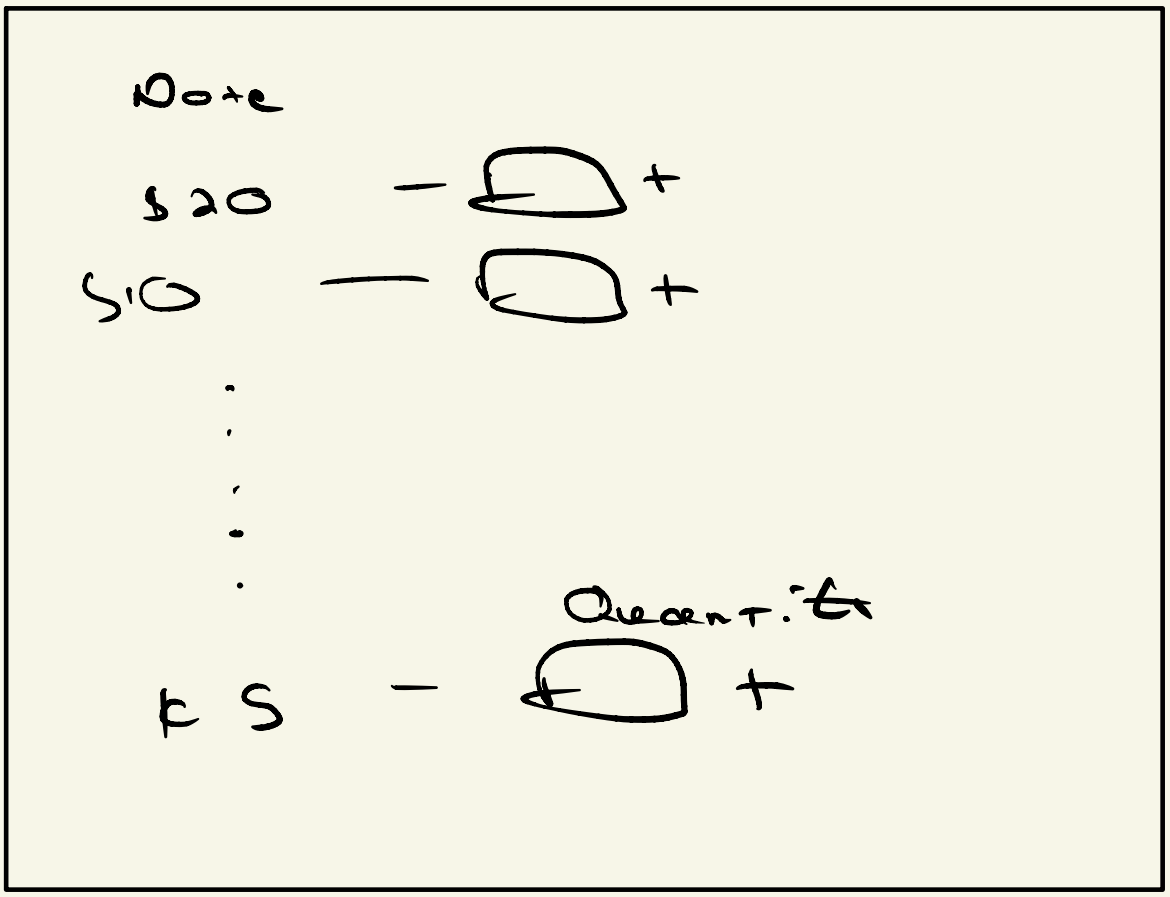
Cashier's

Cashier Default Page

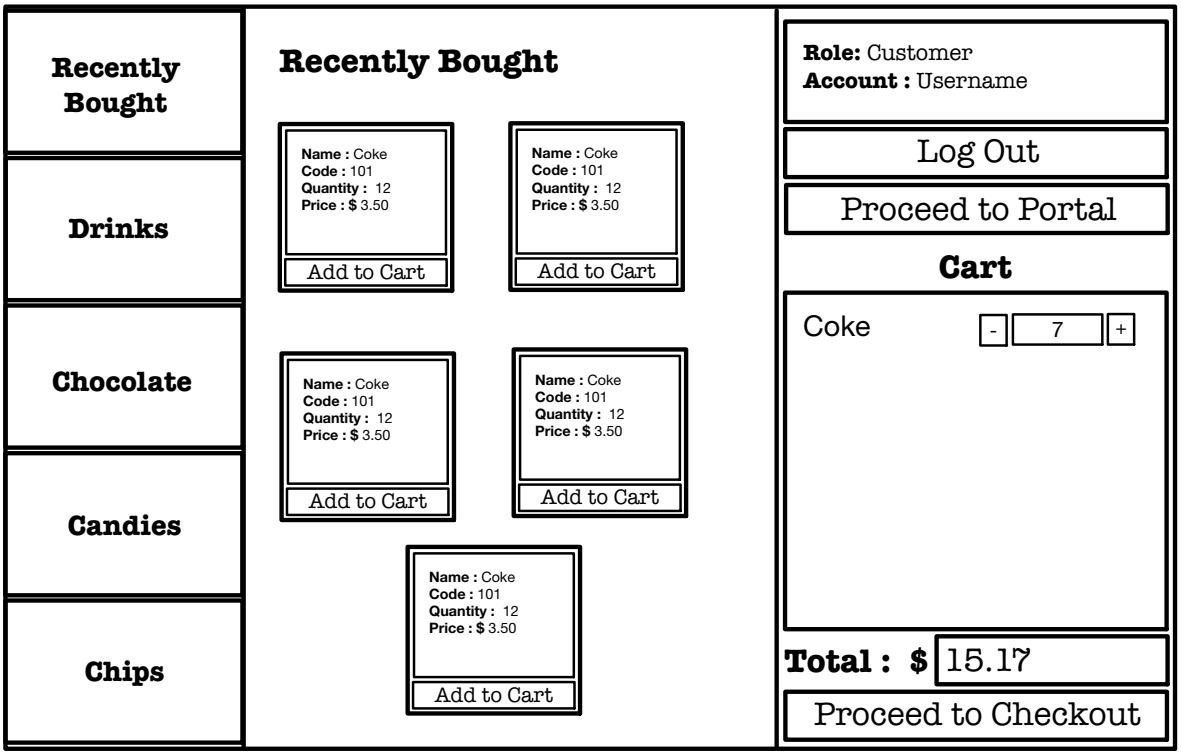
Picture of Product or just details



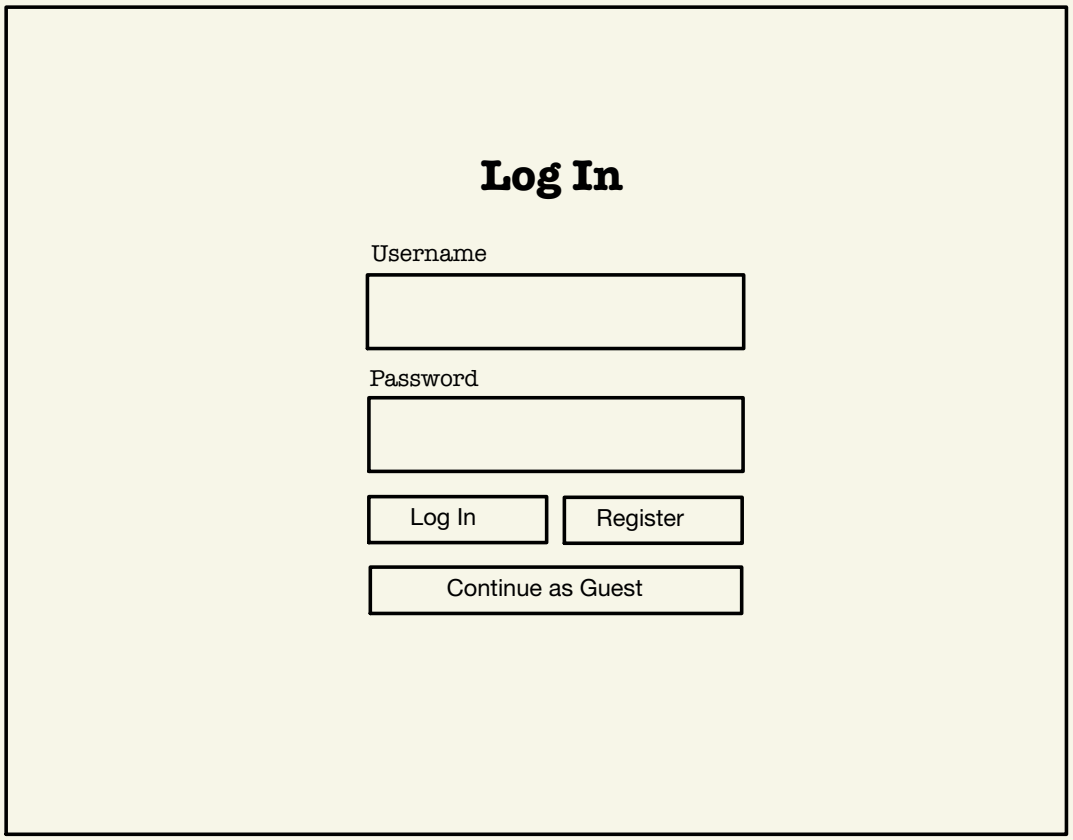

Cashier's Portal
Username
Generate R1
Generate Sales Report
the Cashier Dob
> Pop Up


Note
$20
$10
$5
Quarter

B.2 Hi-fi user stories

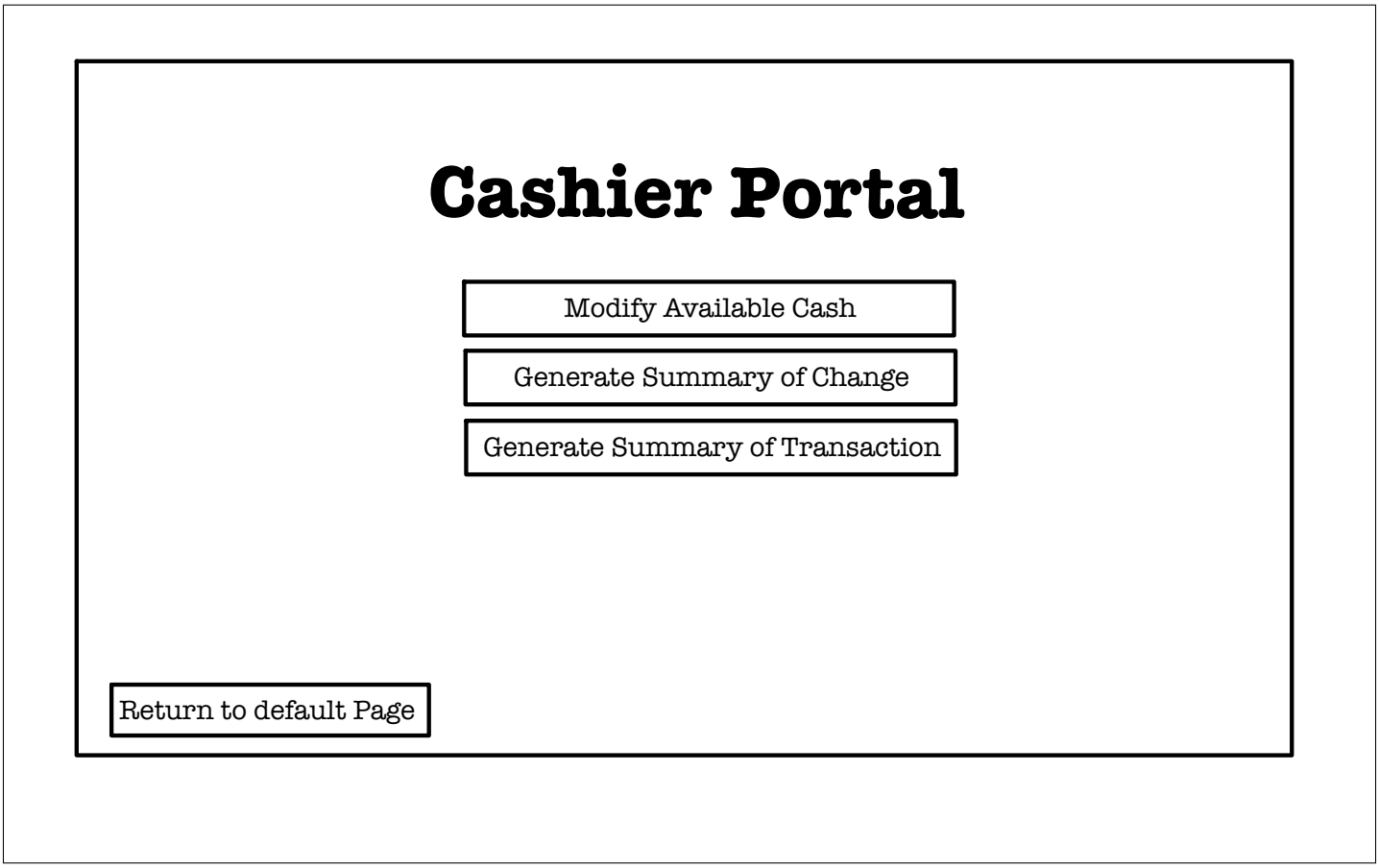

Recently Bought
Drinks
Chocolate
Candies
Chips
Recently Bought
Name : Coke
Code : 101
Quantity : 12
Price : $ 3.50
Add to Cart
Name : Coke
Code : 101
Quantity : 12
Price : $ 3.50
Add to Cart
Name : Coke
Code : 101
Quantity : 12
Price : $ 3.50
Add to Cart
Name : Coke
Code : 101
Quantity : 12
Price : $ 3.50
Add to Cart
Name : Coke
Code : 101
Quantity : 12
Price : $ 3.50
Add to Cart
Role: Customer
Account : Username
Log Out
Proceed to Portal
Cart
Coke
-
7
+
Total : $
15.17
Proceed to Checkout

The Log In Page



Register Page

How Error message should be displayed


Cashier Portal
Modify Available Cash
Generate Summary of Change
Generate Summary of Transaction
Return to default Page

Seller Portal

Manage items

Generate List of Available items

Generate Summary

Return to default Page

Owner Portal

Access Seller's Portal

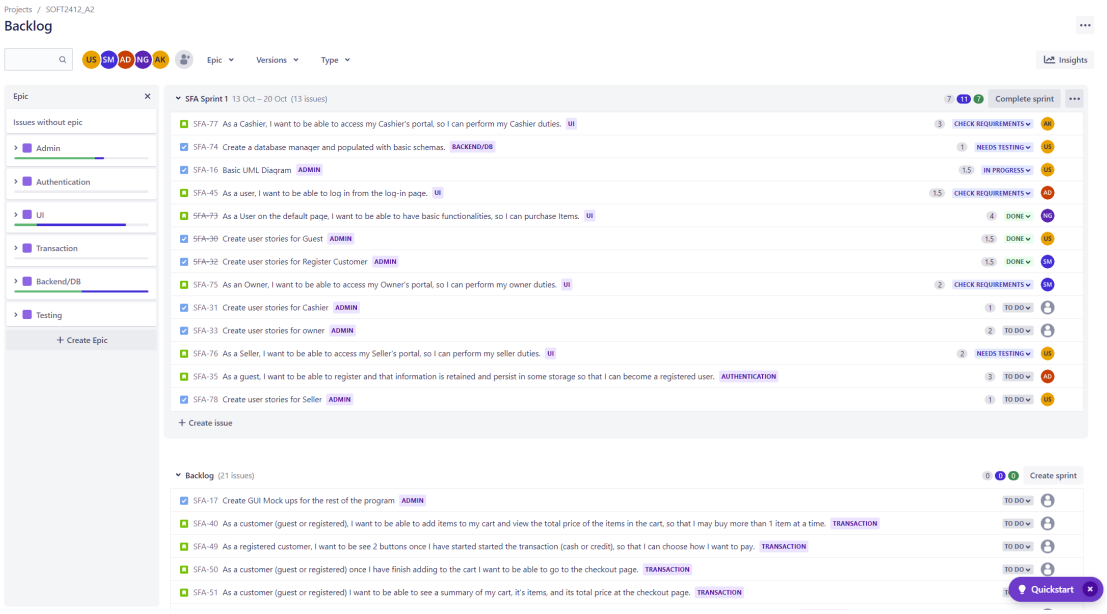
Access Cashier's Portal

Manage Privileged Users

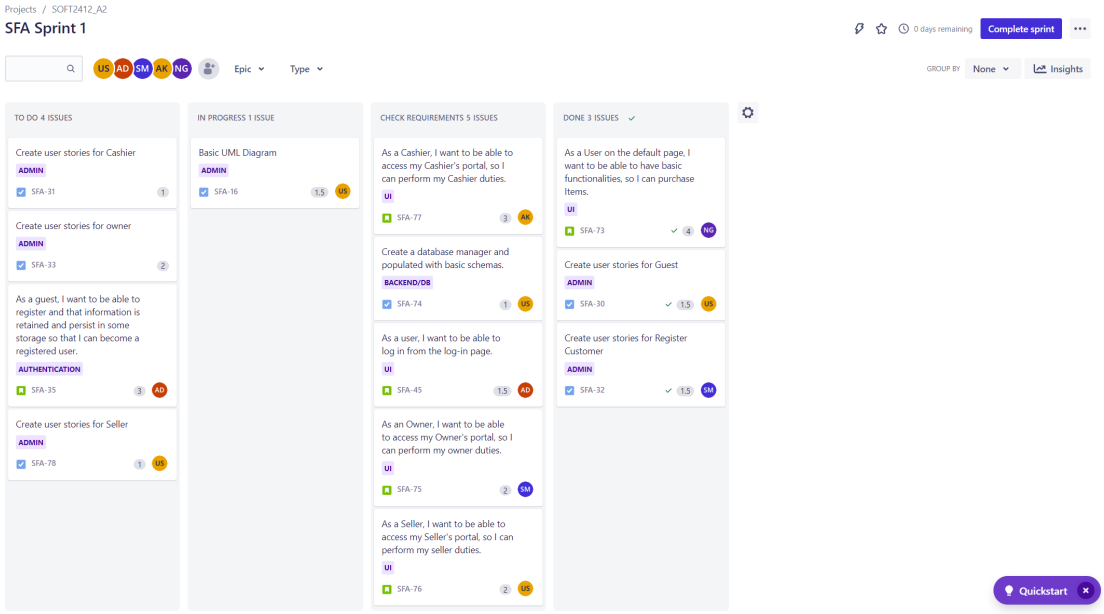
Generate User Summary

View Cancelled Transaction

Return to default Page



(a) Scrum Board



(b) Kanban Board

Figure 4: Comparison between Scrum Board and Kanban Board

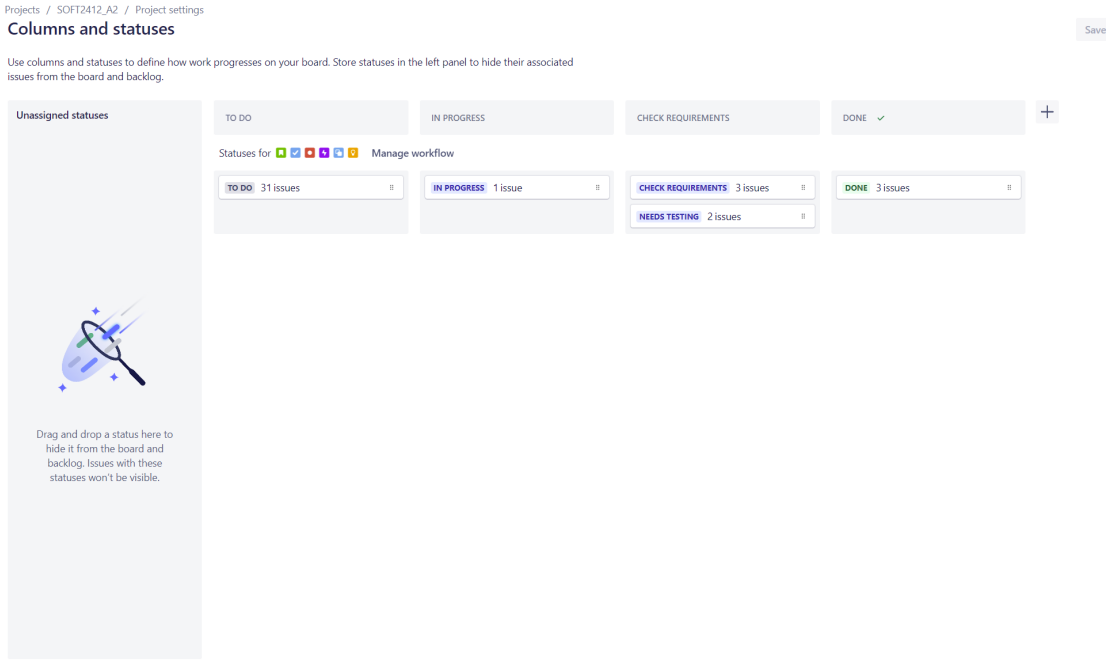


Figure 5: The Scrum Kanban board

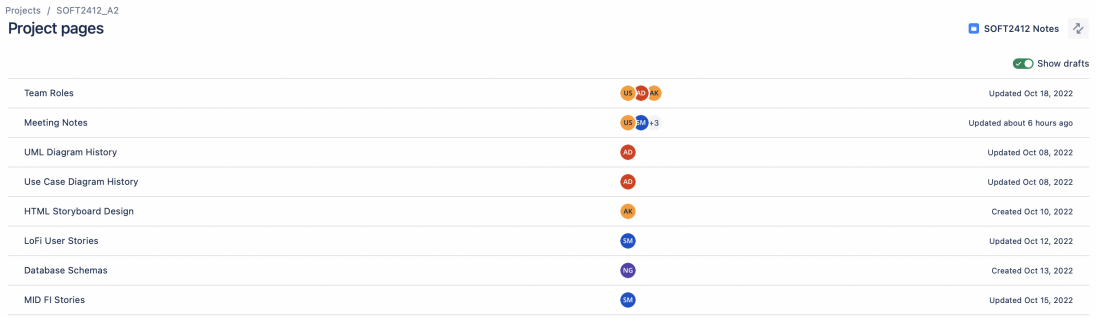
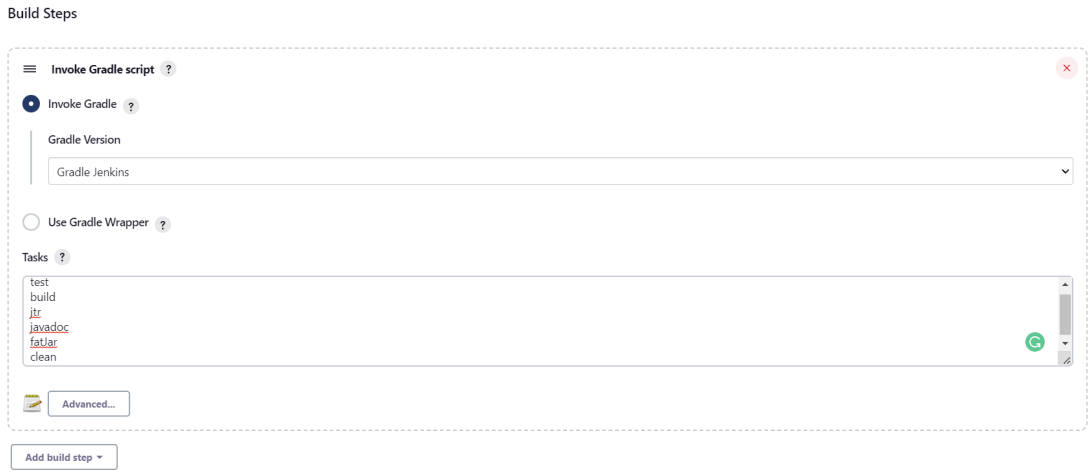
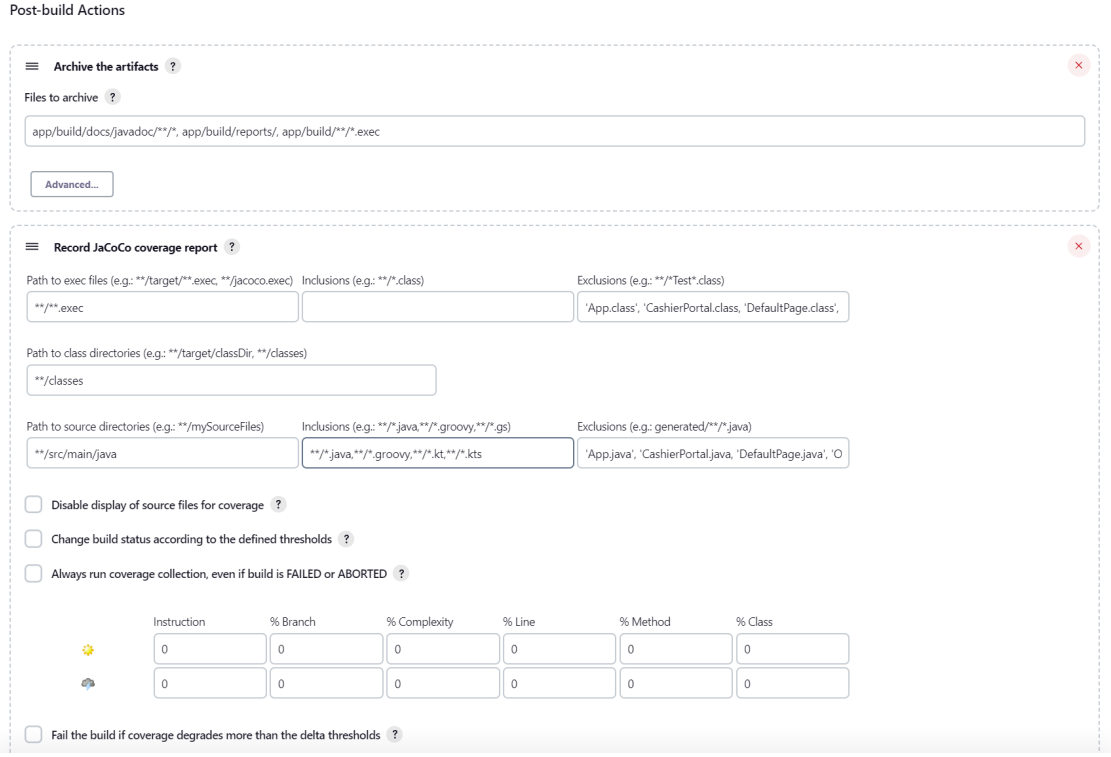


Figure 6: Screenshot of the 'Project Pages' tab, which highlights all the confluence pages used for documentation



(a) Build Tasks



(b) Post Build

Figure 7: Build and Post Build Actions

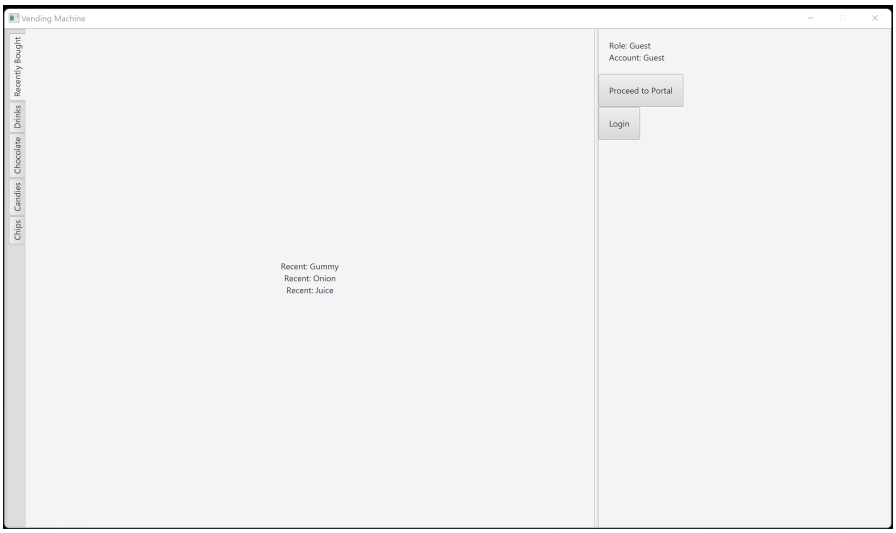


Figure 8: Caption

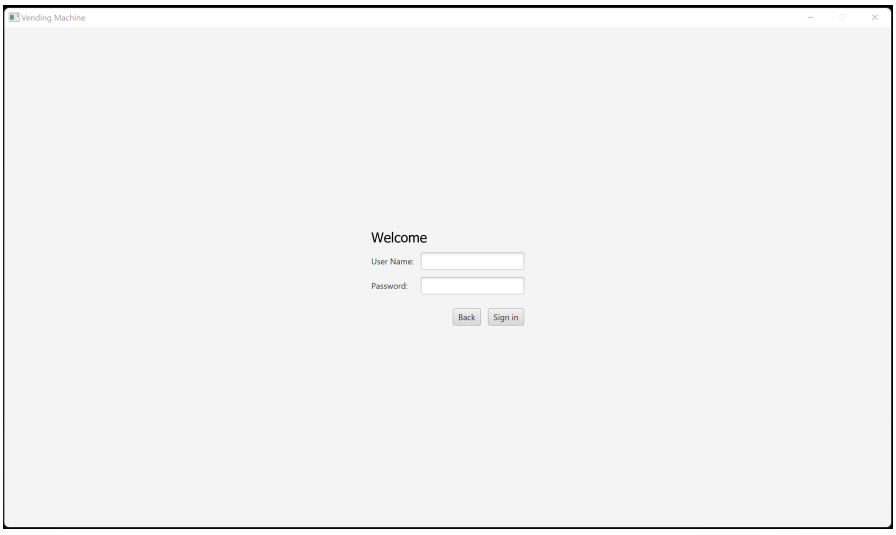


Figure 9: Caption

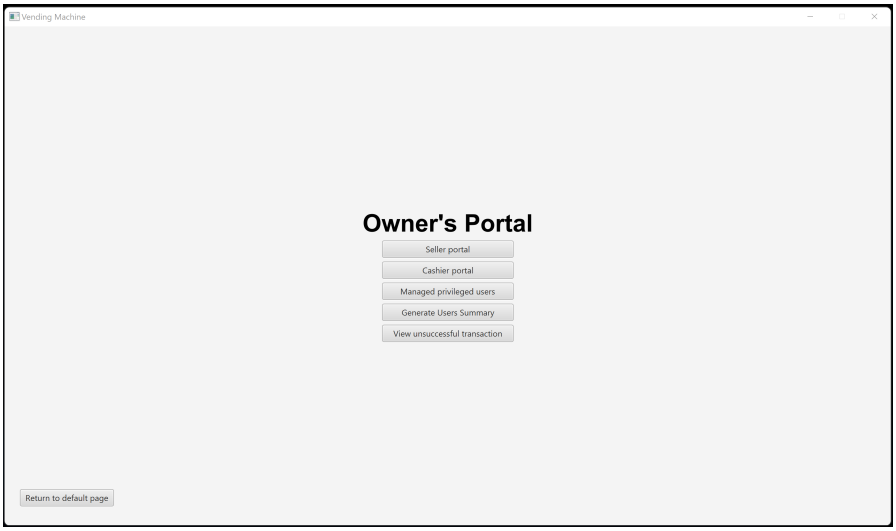


Figure 10: Caption

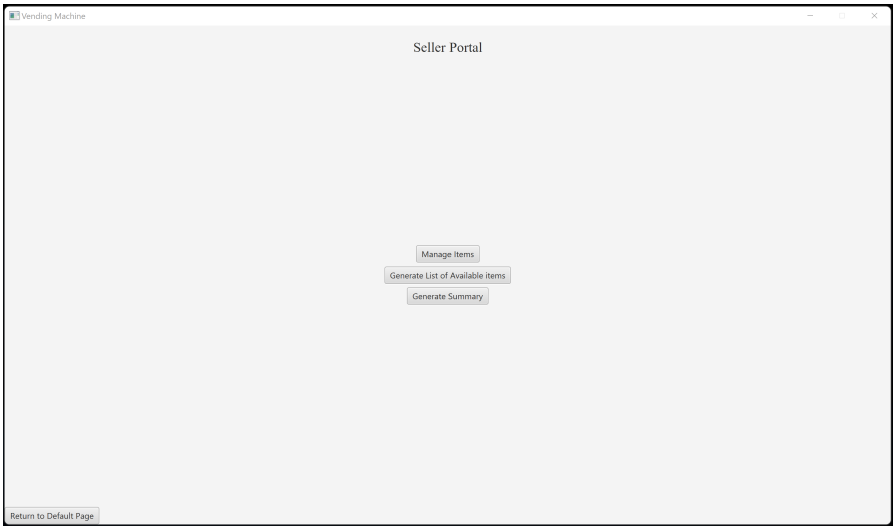


Figure 11: Caption

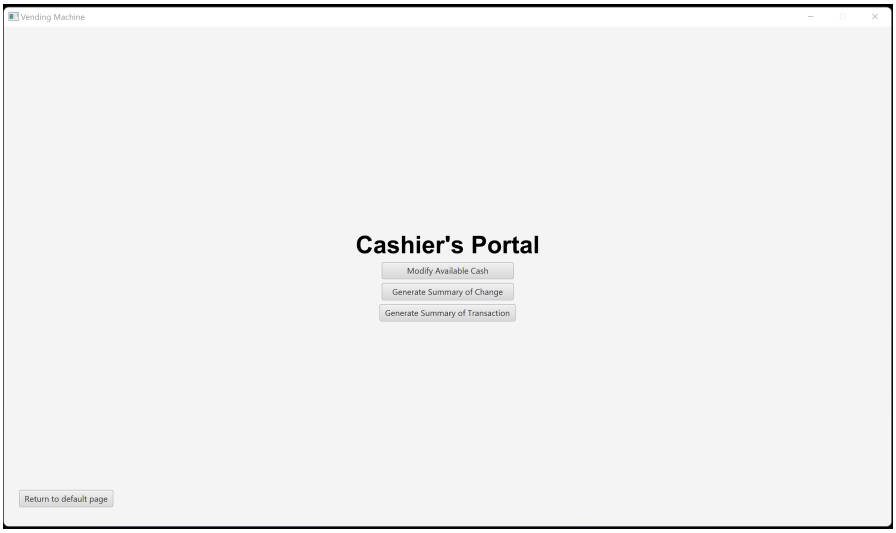


Figure 12: Caption

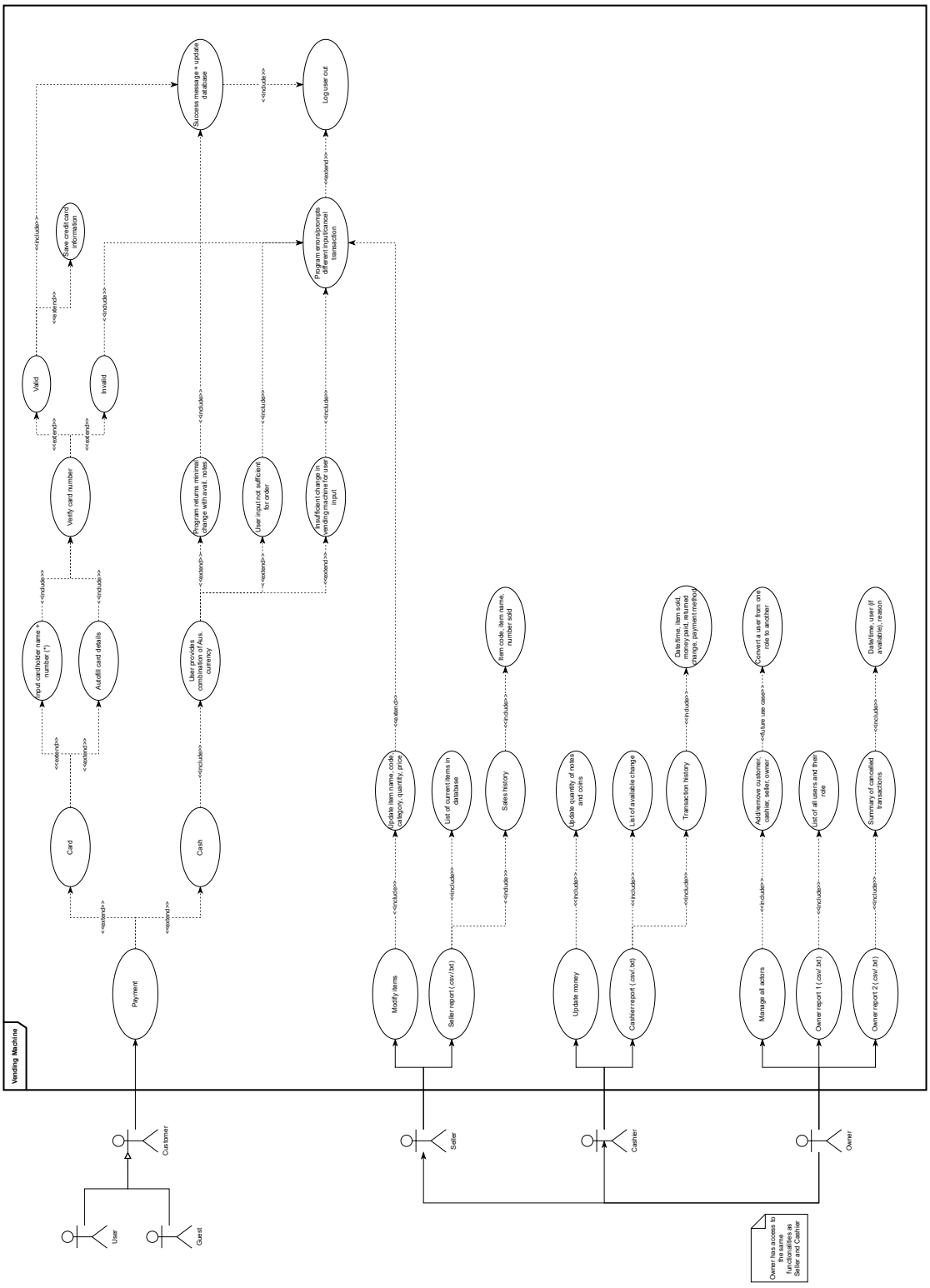


Figure 13: Use case diagram breaking down the project requirements.