CSC 648/848: Software Engineering

RecoveryNote

Everything you need for school in one place

Team 06

Joel Giannelli
Hongjie Li
Tamir Rasheed
Ivan Cebreros
Kuncheng Wu
Nyambayar Purevtseren

Milestone 2

Revision ID	Revision Date
Milestone 2 v1	**10/20/21**

Objective

1. Data Definitions V2

A-1: Data Definitions V2

Name	Definition (& Examples)	Usage
Account	Account that utilizes application and increases functionality.	Users can access their accounts to save account specific information and utilize application features.
School	School that the user would like to see listings from.	Users can select their school they would like to browse listings from and make/create listings.
Registered User	A person who signed up with an account on the application, e.g., Mike, @mike100, school info, age, gender, address	This information is collected so that users can interact with one another, and have the supplied information at hand easily.
User Preferences	Preferences per account, these can include search settings, filtering, and user specific options.	User preferences will be used to provide users a personalized experience based on the preferences set.
Listings	A listing that contains user specified images, description, and product based on the item they're trying to sell.	The user can view listings that other users have created, or can create their own listing with a product they wish to sell.

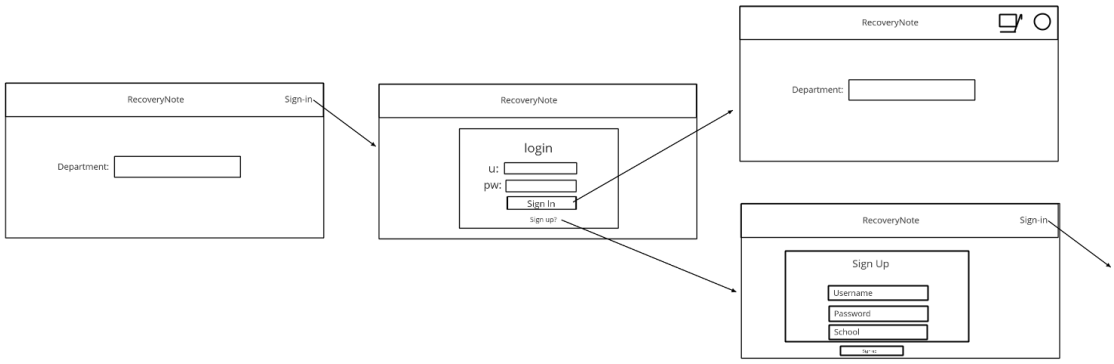
A-2: Functional Requirements V2

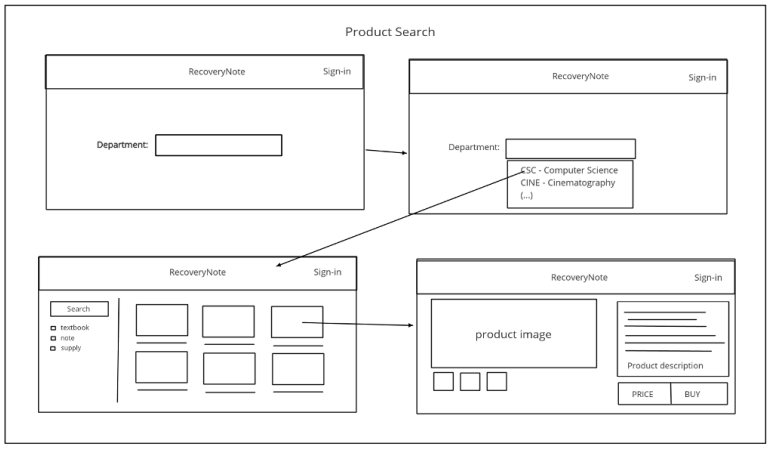
1. A store is a website where account users can buy and sell school supplies.
2. Users can create a unique account
3. Customers with an account can log in with a unique account
4. Account users with an account can log out of the account
5. Account users with a unique user account can list which school they attend and what classes they are taking so that the store will show supplies (textbooks, notes, etc.) that would relate most to that account users
6. Users who do not have a user account can browse the store and see what supplies are available. However, to make a final purchase, they would need to make a unique user account.
7. Account users have a shopping cart.
8. A shopping cart is to be filled with items that the user wants to buy
9. Account users can fill up a shopping cart with supplies they want to purchase
10. Account users shall put in information needed to finalize purchases. This information includes the payment method and the address that the items that will be sent.
11. Account users can choose to save information for future purchases.
12. Account users can leave reviews on items that are sold multiple times.
13. Account users can put up items to sell on the website.
14. Account users who put items to sell on the store shall put a title of the item, a description of the item, when the item was previously used (or if never used), and a price of the item

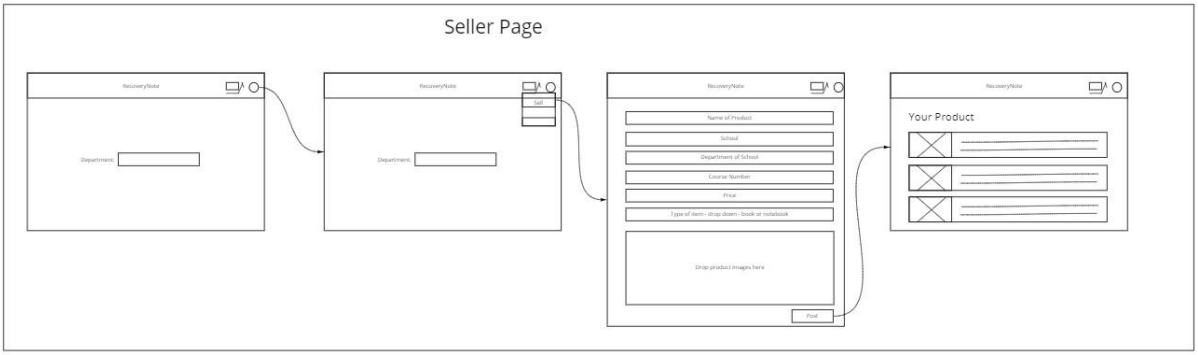
---

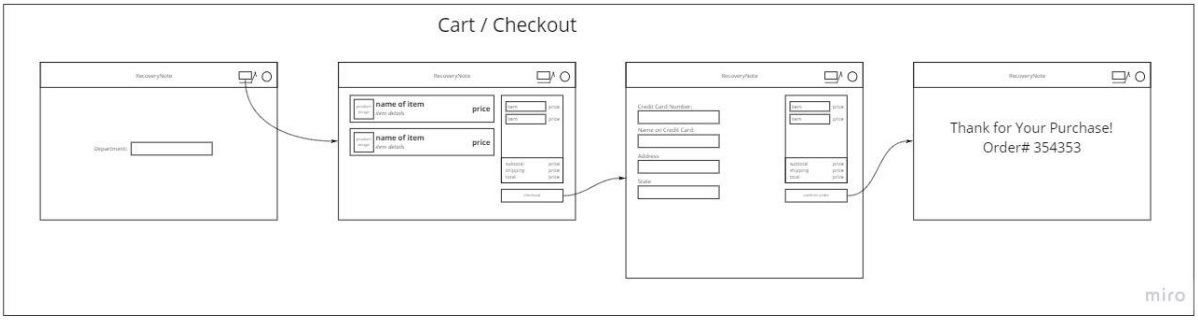
A-3: UI Mockups and Storyboards

Mockup:

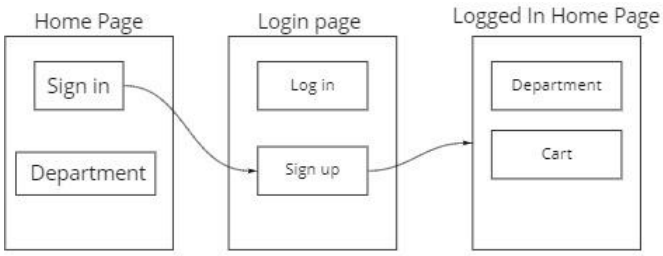

RecoveryNote
Sign-in
Department:
RecoveryNote
login
u:
pw:
Sign In
Sign up?
RecoveryNote
Department:
RecoveryNote
Sign-in
Sign Up
Username
Password
School

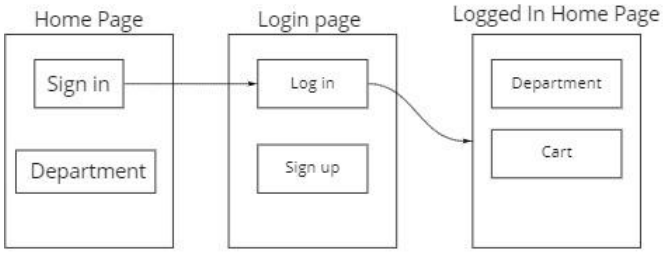

Product Search
RecoveryNote
Sign-in
Department:
RecoveryNote
Sign-in
Department:
CSC - Computer Science
CINE - Cinematography
(...)
RecoveryNote
Sign-in
Search
textbook
note
supply
RecoveryNote
Sign-in
product image
Product description
PRICE
BUY

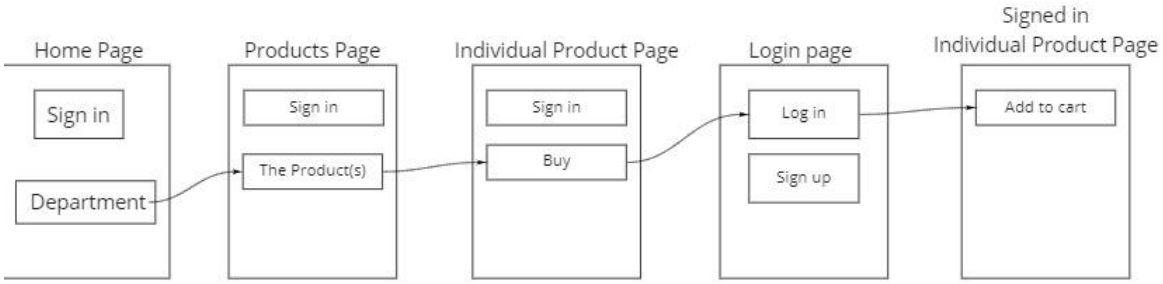

Seller Page
Your Product

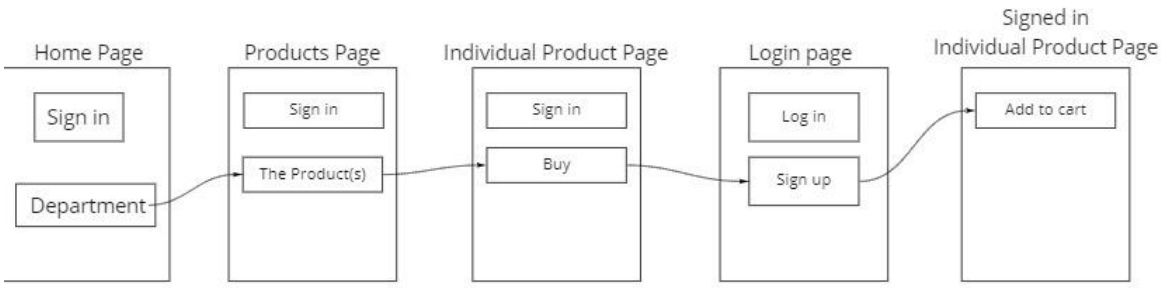

Cart / Checkout
name of item
price
Thank for Your Purchase!
Order# 354353
miro

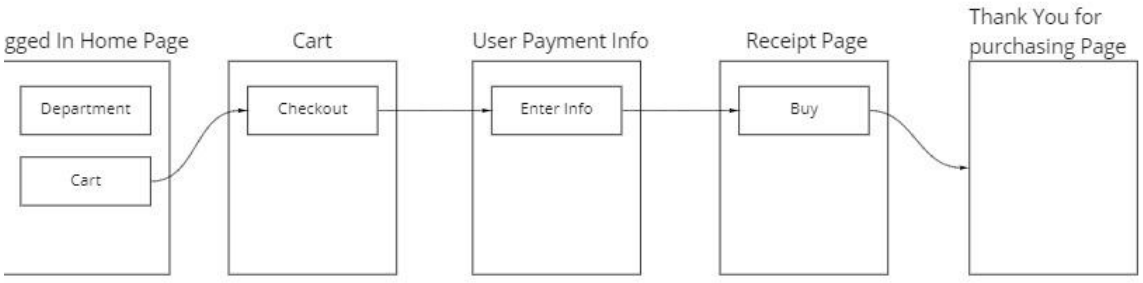
Storyboard:


Home Page
Sign in
Department
Login page
Log in
Sign up
Logged In Home Page
Department
Cart


Home Page
Sign in
Department
Login page
Log in
Sign up
Logged In Home Page
Department
Cart


Home Page
Sign in
Department
Products Page
Sign in
The Product(s)
Individual Product Page
Sign in
Buy
Login page
Log in
Sign up
Signed in
Individual Product Page
Add to cart


Home Page
Sign in
Department
Products Page
Sign in
The Product(s)
Individual Product Page
Sign in
Buy
Login page
Log in
Sign up
Signed in
Individual Product Page
Add to cart

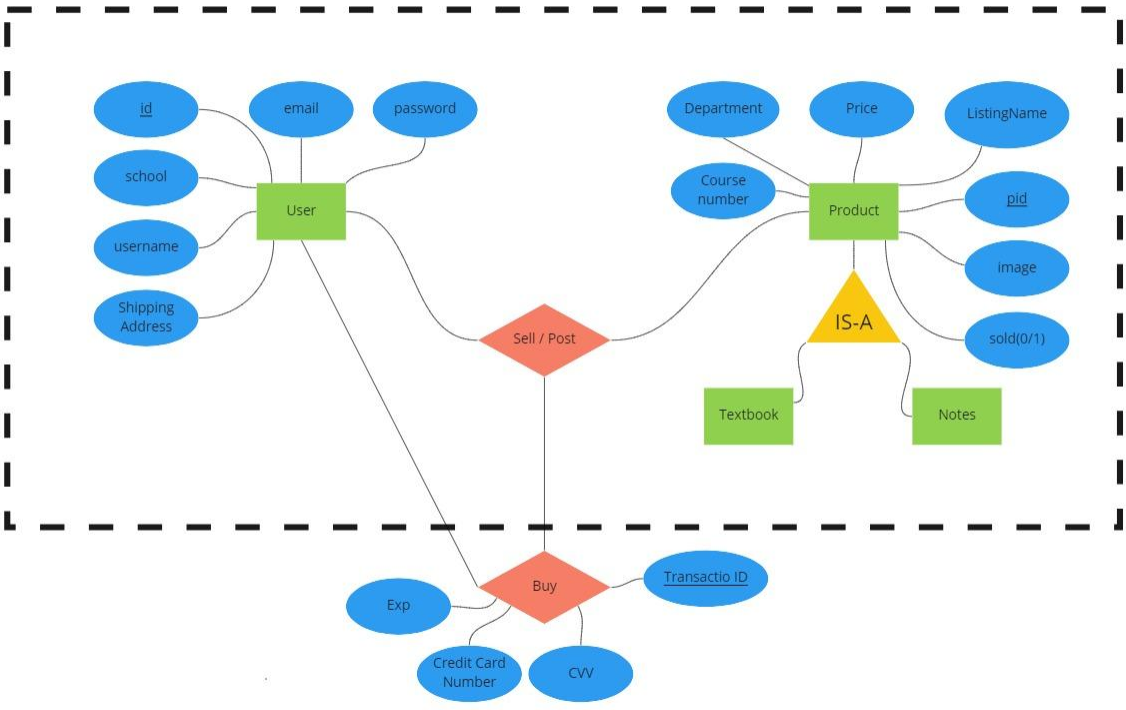

gged In Home Page
Department
Cart
Cart
Checkout
User Payment Info
Enter Info
Receipt Page
Buy
Thank You for
purchasing Page

4. High level Architecture, Database Organization

APIs

A-4: High level Architecture, Database Organization

Entity Relation Diagram:



Schema:

User					
id	school	username	email	address	password
int	nvarchar	nvarchar	nvarchar	nvarchar	varchar
Primary Key					

Product / Sell						
id	department	price	listing_name	image_url	sold	seller_id
int	varchar	float	varchar		boolean	int
Primary Key						Foregin Key

Buy / Transaction Records					
transaction_id	buyer_user_id	pid	credit_card	exp	cvv
int	int	int	int	int	int
Primary Key	Foreign Key				

Add/Delete/Search architecture	Functional Requirement
Search/Display products	When users browse
Add/Delete/Search for Users	When users register
Add product	When user sell a product
Search for product	When user buy a product

APIs we plan to use

Login: Logging into account to access relevant information.
route: /login

Signup:
route: /signup

Post/Sell product:
route: /product/post

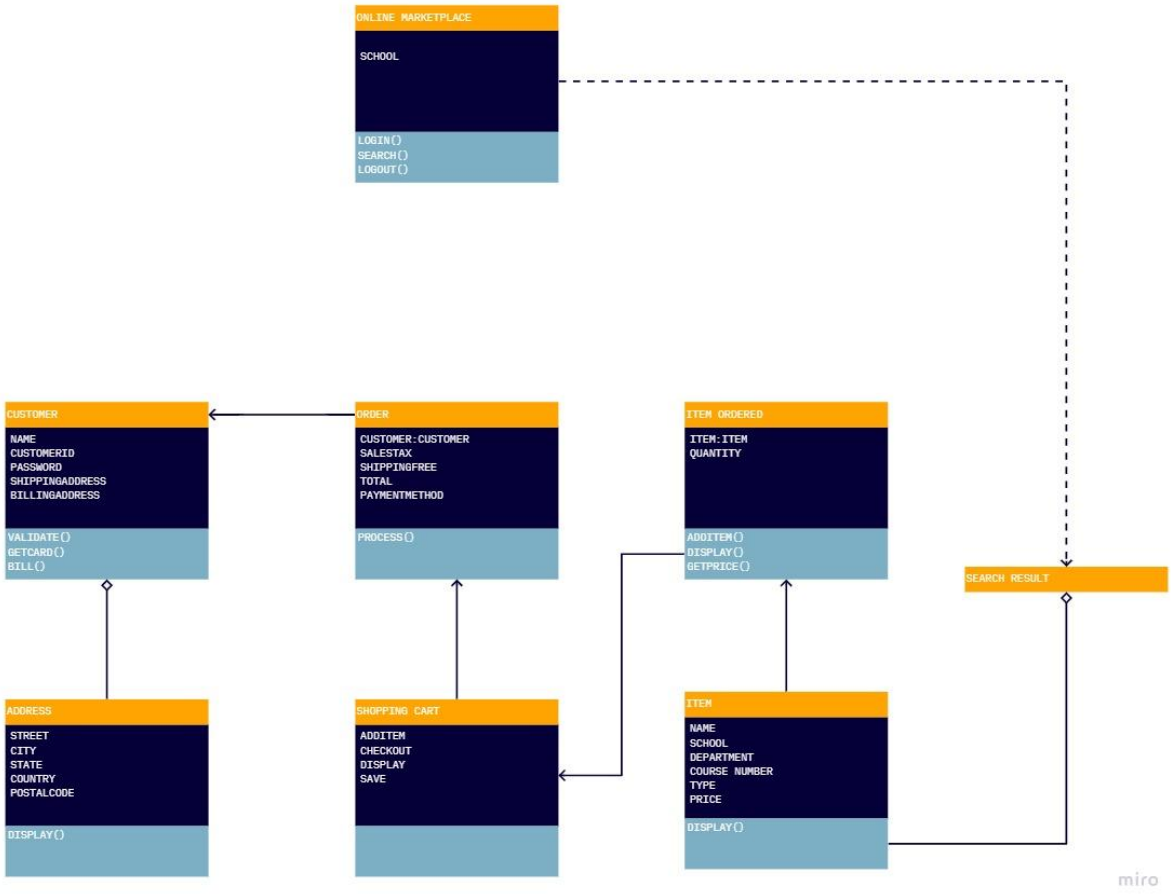
Buy Product:
route: /product/buy

Back End:

We are currently planning to use Express framework.

5. High Level UML Diagrams

A-5: High Level UML Diagrams



---

6 .Identify actual key risks for your project at this time

A-6: Identify actual key risks for your project at this time

Skill Risks and Mitigation Plan:

- When studying a new tool/skill, there is a risk that it may take longer than previously expected.
  To resolve this we:
  - Should start early
  - Examine other resources and options
  - Work together as a group on it to speed up the process
- If there are delays due to personal matters, then the overall progress and plan is affected. This affects development/implementation time.
  To resolve this we:
  - Should start early
  - Plan ahead
  - Communicate with group members
- If a group member (or multiple) are stuck on a particular problem/implementation/bug, this can cause delays to the overall progress and timeline of development.
  To resolve this we:
  - Communicate to our group mates(having a different perspective is important in understanding and solving a problem)
  - Be sure to thoroughly read documentation, as certain key aspects (even small particular details) can be crucial when debugging/solving a problem
  - Take a step back and evaluate other aspects of the project, not just the current issue/bug

Schedule Risks:

- In the event that a group member has to change their plans (either availability, aspects of the project, etc) and does **not** communicate with the group can and usually will lead to issues like not being able to meet a deadline, cause integration errors, work overlapping and uneven work spread, among other things.
  To resolve this we:
  - Provide constant communication on **anything** and **everything**.
  - Maintain code and implementations carefully, as the merging of two features can cause conflict.

Teamwork Risks:

- If multiple members miscommunicate on which section/feature to work on, it would be a waste of time and reduces progress on other features.
  To resolve this we:
    - Communicate properly and frequently
    - Understand tasks and roles throughout each meeting
- If code, files and resources are incorrectly managed it could cause problems for other members who have built their functionality on a certain foundation.
  To resolve this we:
    - Review each others code
    - Remove complexity from certain files once they become too large to efficiently manage
    - Follow code style as agreed upon by the group
- If back-end and front-end members do not provide help and understanding of others respective work, functionalities between the two will become difficult to complete. This may also lead to a lack of implemented functionalities due to the lack of comprehension from both sides.
  To resolve this we:
    - Ask questions and be proactive about learning
    - Improve understanding and explain back-end/front-end to group in meetings

---

7. Project management

A-7: Project management

Our team managed the M2 milestone by working together on the major product defining tasks and then dividing up the smaller tasks that branched from the main questions. The team as a whole collaborated and contributed to the design and layout of the overall website by providing input and simultaneously working on the high level architecture of our database, the mock ups and storyboards. While it may have been slower to have the entire team work on these individual pieces, we felt it to be important that every team member be on the same page regarding the infrastructure of the website regarding both the front end and the back end. In ensuring that every team member was well versed in both we can guarantee that every member is working towards the same goal(s) when working on their individual components.

In order to track each member's progress our scrum master used trello to monitor everyone's progress and ensure that everyone was projected to complete their portion in a timely manner. In our group meetings, we recapped our overall goals and checked in as to how everyone was doing with their individual tasks. Speaking to transparency, the trello is accessible to all team members so every group member can see their own progress in relation to the rest of the group.

Overall, our entire team was very proactive in meeting deadlines in a timely fashion. Our group work at the start of the milestone made it very easy for everyone to understand the vision and goals of this project. In setting clear expectations for our final product we were all able to work towards these expectations. Our approach to this milestone, combined with trello, ensured that our team had clear expectations for each other and that these expectations were ultimately met in the final milestone.