Alexander Reese

Yan Huang

SWE 6623

Final Project Report

Idea

I wanted to create a software that can help user get a better control over their buying and security. The importance of the software is for anyone who can be forgetful or is compulsively buying items. It can also help people who wants extra security for their purchases and help manage what items they own and want to purchase. This software can be used to help users save money find or keep track of old product they probably own that is held in their storage. I wanted to create a solution that can help people not only save money and time, but also add extra security when it comes to online purchases through the web.

Plan& Manage

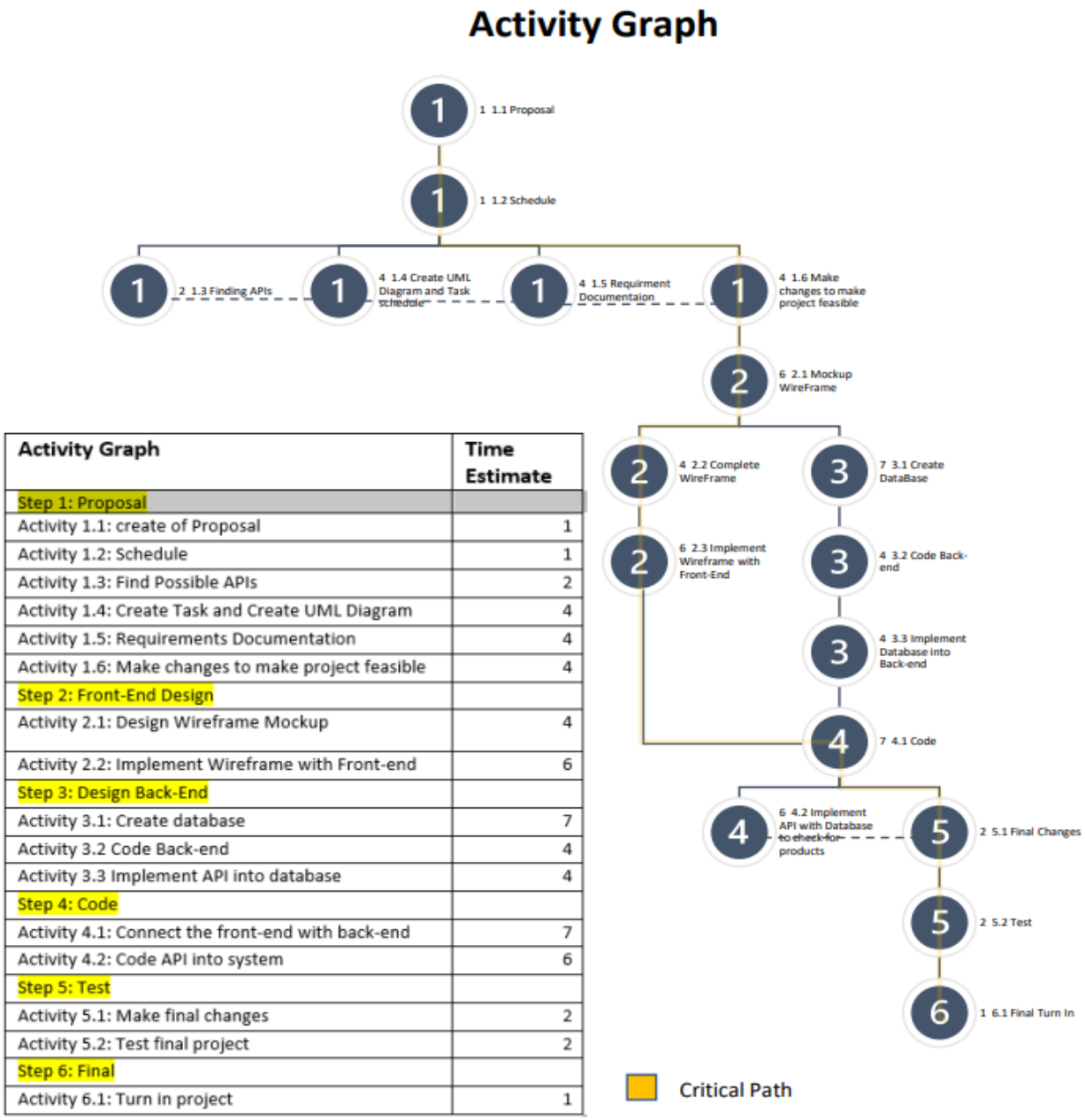
In terms of the planning phase, I started this project in the beginning of the semester and planned to finish it by the end of the semester. The first step was all about how the project was going to run, what language was going to be used to run the program and finding the requirements for the software. When it came to scheduling, I created an estimated time on how long the project will take and when it will be completed.

Schedule

Phase 1: Pre-Planning	Estimate (days)	Phase 2: Project Design	Estimate (days)	Phase 3: Project System Level Test	Estimate (days)	Phase 4: Final Project Deliverable	Estimate (days)
Step 1.1 Create Proposal		Step 2.1 Design		Step 3.1 Testing		Step 4.1 Turn in	
Activity 1.1.1: Find importance of project	1	Activity 2.1.1: Mockup Wireframe	6	Activity 3.1.1: Make necessary changes	2	Activity 4.1.1: Turn in Project/Present	1
Activity 1.1.2: Find possible API	1	Activity 2.1.2: Final Wireframe	4	Activity 3.1.2: Test	2		
Activity 1.1.3: Estimate Milestones/Completeness	2	Activity 2.1.3: Implement Wireframe with Front-End	6	Activity 3.1.3: Test Final Project	1		
Step 1.2 Create Schedule and Diagrams		Step 2.2 Database					
Activity 1.2.1: Find possible task/activities	4	Activity 2.2.1: find application to store Data	3				
Activity 1.2.2: Create Task Schedule	4	Activity 2.2.2: Create Database	7				
Activity 1.2.4: Create UML Diagram	4	Activity 2.2.3: implement Database into Back-end	4				
Step 1.3 Create Requirements Documentation		Step 2.3 Code					
Activity 1.3.1: List non-functional requirements and functional requirements	4	Activity 2.3.1: Learn/Find program to code project	3				
Activity 1.3.2: List requirements and receive feedback	4	Activity 2.3.2: Code to make project functional	6				
Activity 1.3.4: Make changes to make project feasible	4					Project Estimate Completion	73 (days)

September 17 - November 30 = 75 days

Fig 1.



Activity Graph	Time Estimate
Step 1: Proposal	
Activity 1.1: create of Proposal	1
Activity 1.2: Schedule	1
Activity 1.3: Find Possible APIs	2
Activity 1.4: Create Task and Create UML Diagram	4
Activity 1.5: Requirements Documentation	4
Activity 1.6: Make changes to make project feasible	4
Step 2: Front-End Design	
Activity 2.1: Design Wireframe Mockup	4
Activity 2.2: Implement Wireframe with Front-end	6
Step 3: Design Back-End	
Activity 3.1: Create database	7
Activity 3.2 Code Back-end	4
Activity 3.3 Implement API into database	4
Step 4: Code	
Activity 4.1: Connect the front-end with back-end	7
Activity 4.2: Code API into system	6
Step 5: Test	
Activity 5.1: Make final changes	2
Activity 5.2: Test final project	2
Step 6: Final	
Activity 6.1: Turn in project	1

Fig 2.

In figure 1 and 2, this is the schedule and activity graph that shows the estimate amount of days it could take to complete the project. Sadly, there were a lot of setbacks due to other important matters and deadlines. There was also a feasibility issue with the project which was the lack of understanding on how to properly code and program the software. I did not properly scope the project correctly and later realized the difficulty of the project I was building. I managed the project by working on it every week in the semester. A lot of changes have occurred from the development of the project to the later stages of development to make the project easier to build. I manage to change how the product can be stored in a database and to make the simplicity sake for the project, I decided to drop the security portion of the project.

Requirements

The requirements of the proposed project will influence on how the project will run and hopefully prevent failure. The requirement of the project will be listed below

1. The program must be able to take new user accounts
2. The program must be able to take multiple order request
3. The program must be able to run while both child and parent account is active
4. The program must be able to have updatable order details and shipping info.
5. The program must be able to have an updatable database
6. The user must be able to manually interact with their database so they can make edits.
7. The user must be able to interact the gate to stop or proceed with purchases
8. The program must be able to allow users to update their information
9. The program must be able to take orders and send them to the gate in evaluation to go into the database
10.

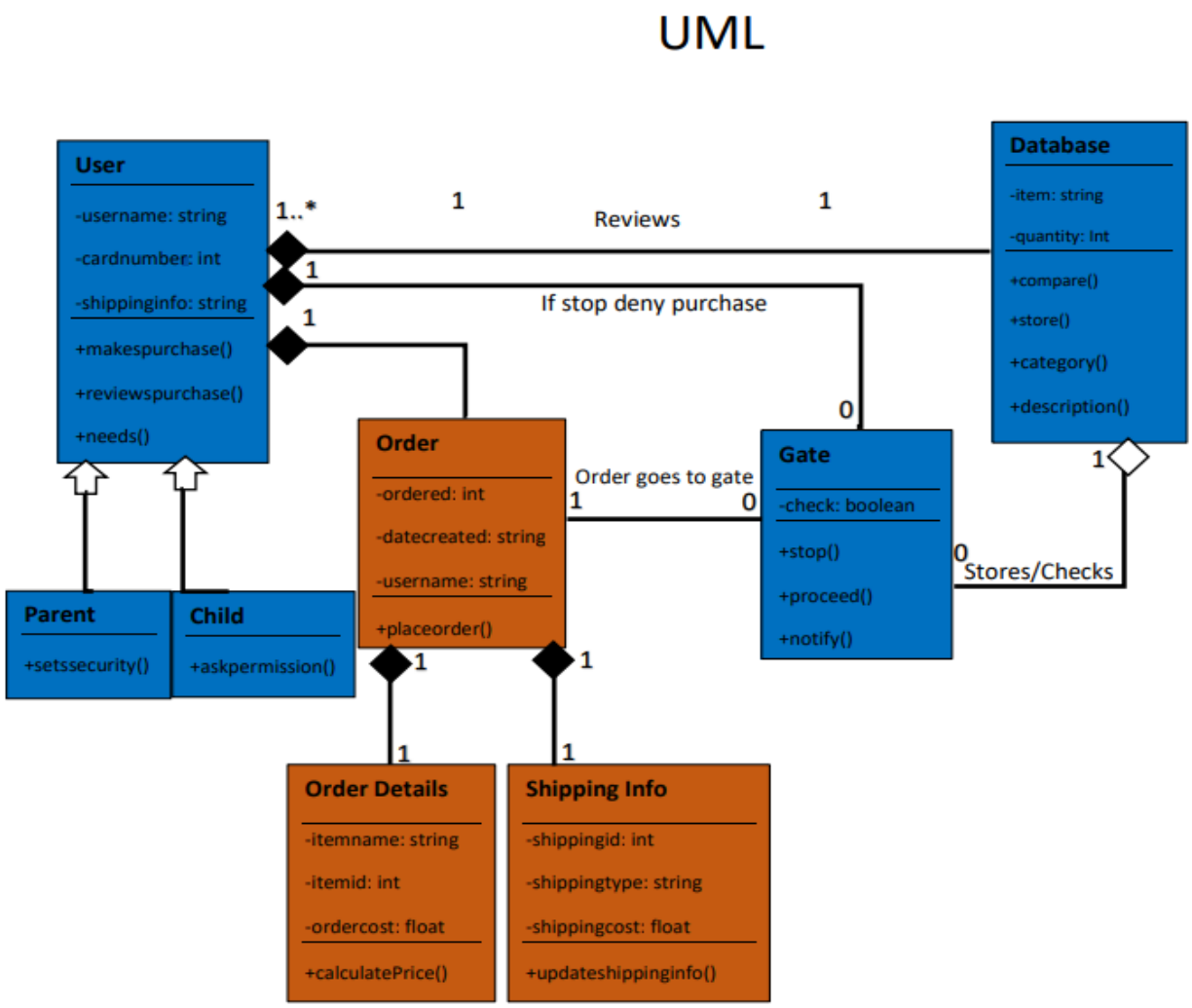


Fig 3.

In figure 3, This is the UML diagram for the proposed project, The UML shows how the classes and objects react to each other with their methods.

Regarding the user interface, the plan was to use Java GUI to be able to interact with the gate and interact with the database.

1. User Interface
    a. a. Pop-up to inform with users about items in user database
    b. b. Navigation to see User's database
    c. c. Yes/No pop-up for approval of purchases
    d. d. Security and notification customization
    e. Navigation to manipulate with the database

Architecture

The architecture style I decided to use was the Layered Style. I felt the layered style was the best style to represent my project as it shows how the project will need to access different tiers to make purchases to enter the product into their database.

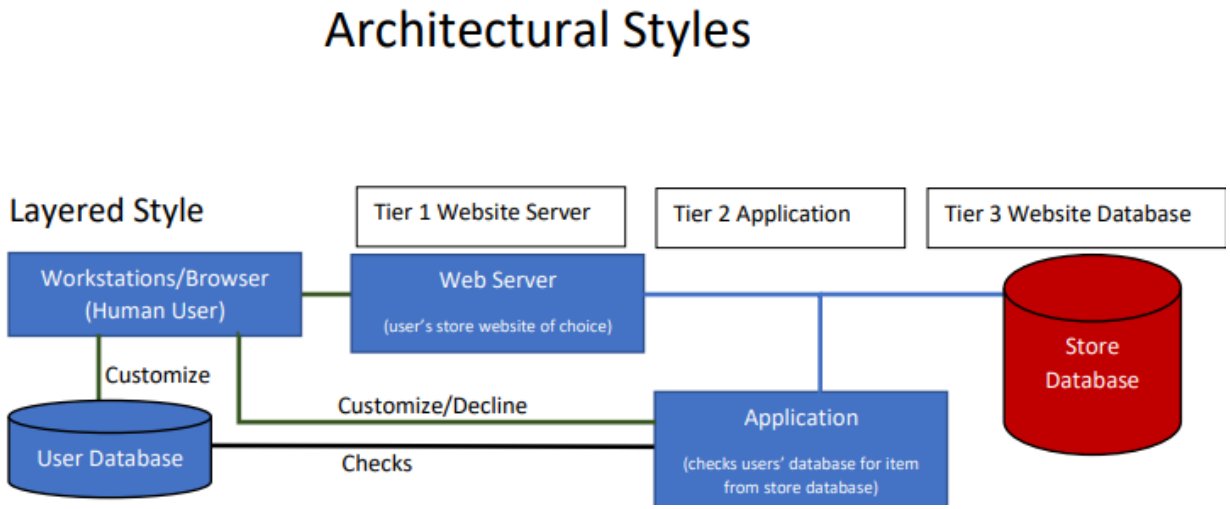


fig 4.

In figure 4, it shows how the layered style plays into action. The user will be given a Database they can customize and manage which is linked to the application. As the user browses the web and tries to make the purchase off the store's inventory, the application will check the user's data base and check for item similar in that category. The user must make an option to proceed or decline the purchase.

Difficulties

There was a lot of difficulties when working on this project. With no background in programming and software development, I understood on the project managing side of things, but when it comes to coding and developing the project, I struggled the most in that criteria. I made a lot of last-minute changes on how the user will be able to access the database and how they can interact with it. When it came to testing the project a lot of the time the file would not update when the user completed their "purchase". I also encountered a lot of loops when it came to allowing the user to choose which

product, they wanted to put into their database which made is more frustrating for me to figure out. I was able to get a couple of the functions and features completed but was not able to complete the project. I learned a lot while building the program and I was close to completing the project, but in the end, I did not get what I wanted to accomplish.

Deploy

I deployed the project to my GitHub (https://github.com/areese30/Final-Project-6623.git), but if the project would to be deployed as a real-world application. It would be used as a third-part application to assist with large database store websites like amazon, Walmart, Target, etc.

Quality of developed project & measurement result

For me to explain the quality of the developed project, if it were completed it would be 6 stars out of 10 if it was on a scale rating system. I would measure it on completeness and if it were able to handle to problem it wanted to handle. I believe it could have handled it better, but due to time constraints and limited knowledge, the quality of the developed project is much lower than the goal.