4/24/2019

ToolDepot

Final Project COP 5339

Lars Koester, Tyler Smith
PROJECT TEAM 6
https://github.com/larskoe/ToolDepot_v2

Table of Contents

1 Introduction

Our goal was to create a tool exchange app that lets users rent out tools to other people in the approximate area. We were ambitious to create a fully functional app with a database and many features. However, we learned that we should take it one by one and we did. We completed all the necessary features first and, in some cases, not the way we originally planned. In the end our app is fully functional for the features specified below. In our struggles we learned hard won lessons that will not soon be forgotten. In the end designing this application was a great learning experience. We hope you enjoy.

2 Proposal

ToolDepot is a tool exchange app for people who do not want to buy the tools they may only need once for a repair at home or for people who have a lot of tools and they want to offer them for rent and make extra money doing so. ToolDepot will allow users to look for tools they need and will be able to contact the owner and ask them to rent them for a specific amount of time. They will also be able to offer their tools for rent to other users. To contact the owner, they will have a build in chat function or an email to contact them. The owner should include basic requirements to rent the tool, such as but not limited to length of rental, cost to rent, how much the tool will be if it damaged and the renter will have to buy the tool, etc.

3 Functional Requirements

The system shall:

- Allow users to make profiles
- Allow users to post items
- Allow users to log in
- Allow users to search for items
- Allow users to view information about items
- Allow the user to view their profile
- Allow user to rent out an item
- Allow user to select an item to rent
- Allow user to view if an item is available

4 Use Cases

1. **Sign Up**
   a. The system prompts user to sign up for ToolDepot
   b. User enters username, full name, email, address, password, city
   c. System confirms successful

2. **Log In**
   a. The system prompts user to enter their credentials
   b. User enters their username, and password
   c. System confirms log in
   d. System takes user to product page

3. **Search Item**
   a. Carry out **Log In**
   b. User selects "Search for Tool" from menu
   c. The user enters desired tool in search bar
   d. System outputs list of possible available items
   e. The user reviews list of items

4. **Check if tool is available**
   a. Carry out Search Item

  b. User selects desired tool and clicks view
  c. User reviews if tool is available
  d. User logs out

5. **Check if tool is available (Variation #1)**
    a. Carry out check if tool is available
    b. User selects rent to rent the tool
    c. User agrees to terms and conditions
    d. User selects rent
    e. User logs out

6. **Rent a tool (as a renter)**
    a. Carry out Search Item
    b. User chooses item from list
    c. User reviews condition and specifications of selected item
    d. User selects proceed to check-out
    e. User schedules return time of item

7. **Rent a tool (as a renter) (Variation #1)**
    a. Carry out **LogIn**
    b. User chooses item from list
    c. User reviews condition and specifications of selected item
    d. User selects proceed to check-out
    e. User schedules return time of item

8. **Post a tool**
    a. Carry out **Log In**
    b. User selects to post tool from menu
    c. User enters the name of category, tool, description, condition, price, and terms of rental
    d. User reviews all entered information
    e. User confirms post

9. **View Profile**
    a. Carry out Login
    b. Select profile from menu
    c. View profile information

5 CRC Cards

Register	
• Registers new users • User enters username, name, password, email address, city	• CustomerDatabase • ItemDatabase

AddItem	
• User enters new item and description • Assigns item to customer	• ItemDatabase • CustomerDatabase

Login	
• Validates username and password	• CustomerDatabase

CustomerDatabase	
• Stores userID, username, name, password, email address, city	

Cart	
• Has items that user wants to rent • Registers item to customer who rented it	• ItemDatabase • CustomerDatabase

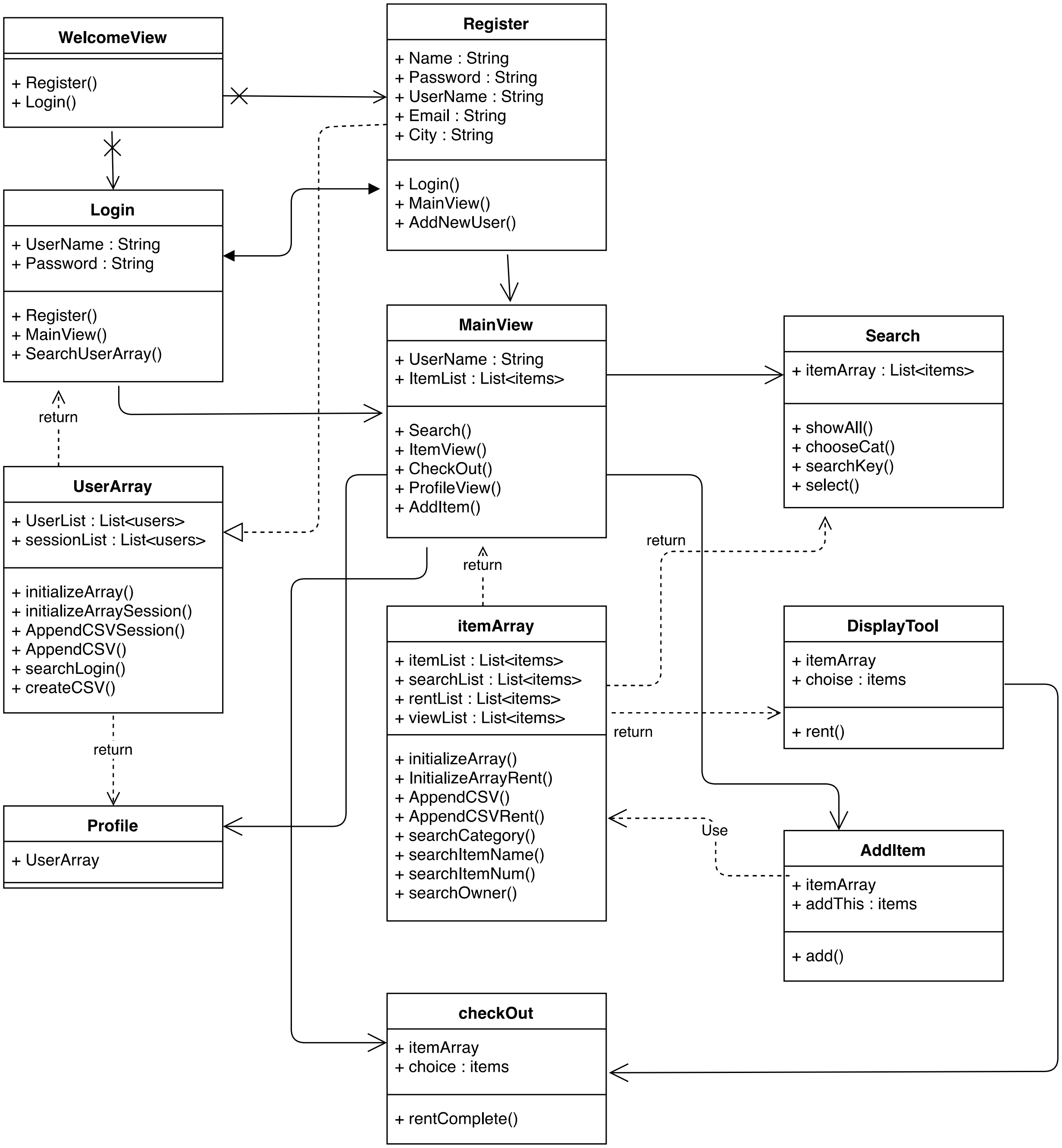
LogOut	
• Terminates user session and logs out	

ViewItem	
• Views Item information such as name, availability, etc.	• ItemDatabase

ItemDatabase	
• Stores userID, itemID, description, image, price, rental period, availability	• Cart • ViewItem

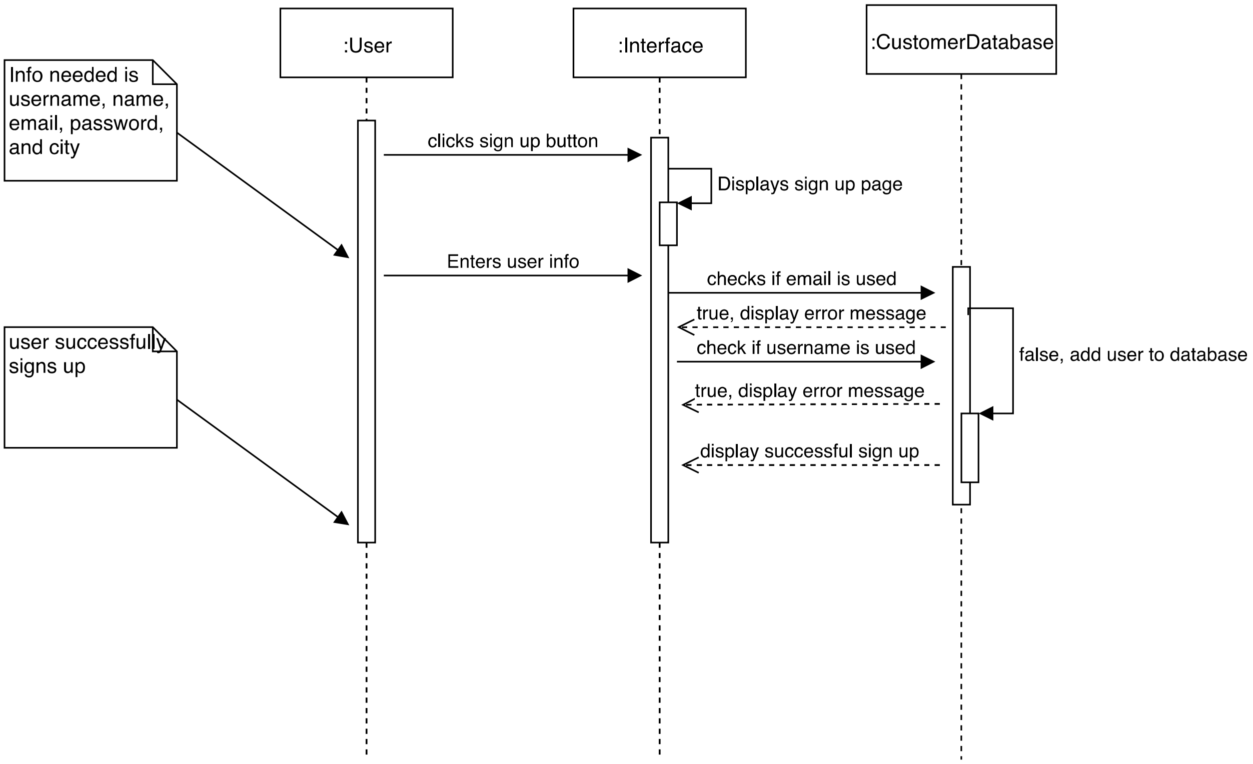
Search	
• Displays all items available for rental • Searches for items	• ItemDatabase

6 UML Class Diagram

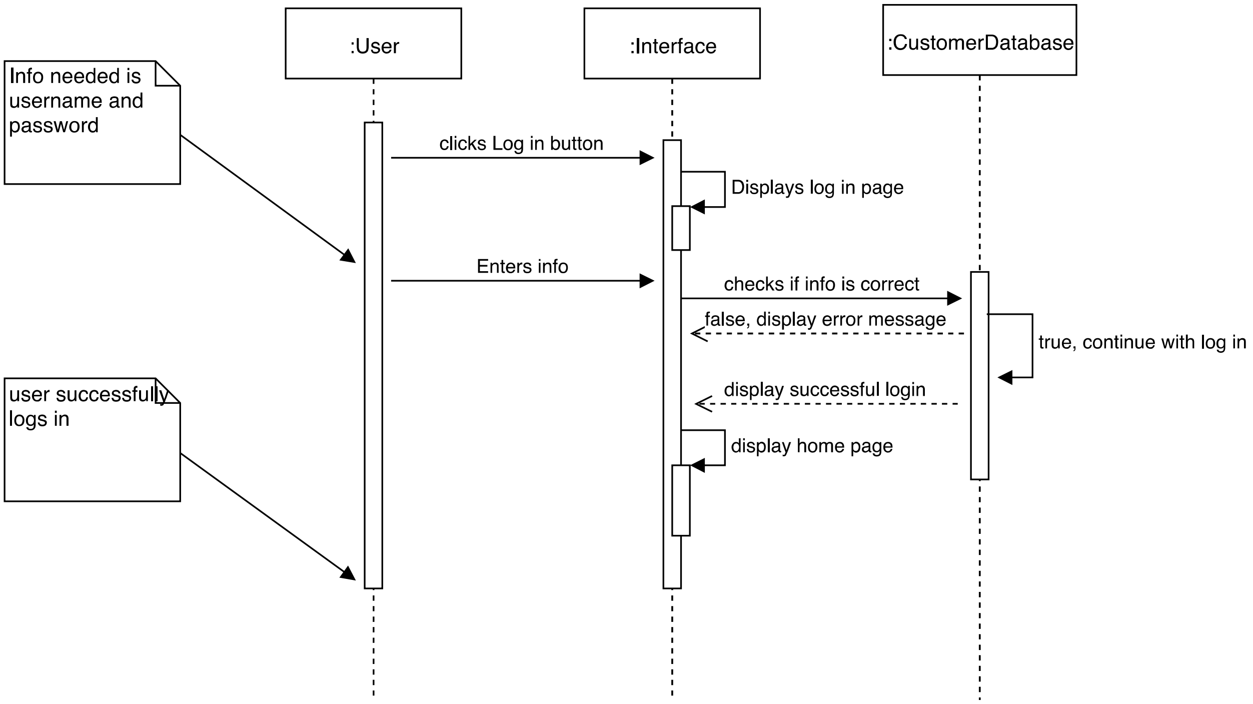


7 Sequence Diagram

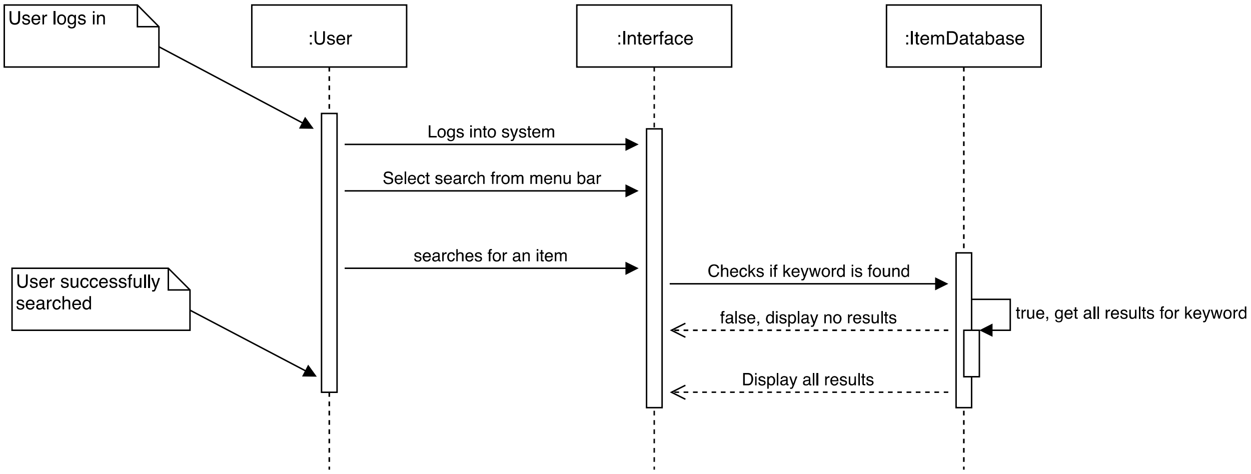
7.1 Sign up


:User
:Interface
:CustomerDatabase
Info needed is username, name, email, password, and city
clicks sign up button
Displays sign up page
Enters user info
checks if email is used
true, display error message
check if username is used
true, display error message
false, add user to database
display successful sign up
user successfully signs up

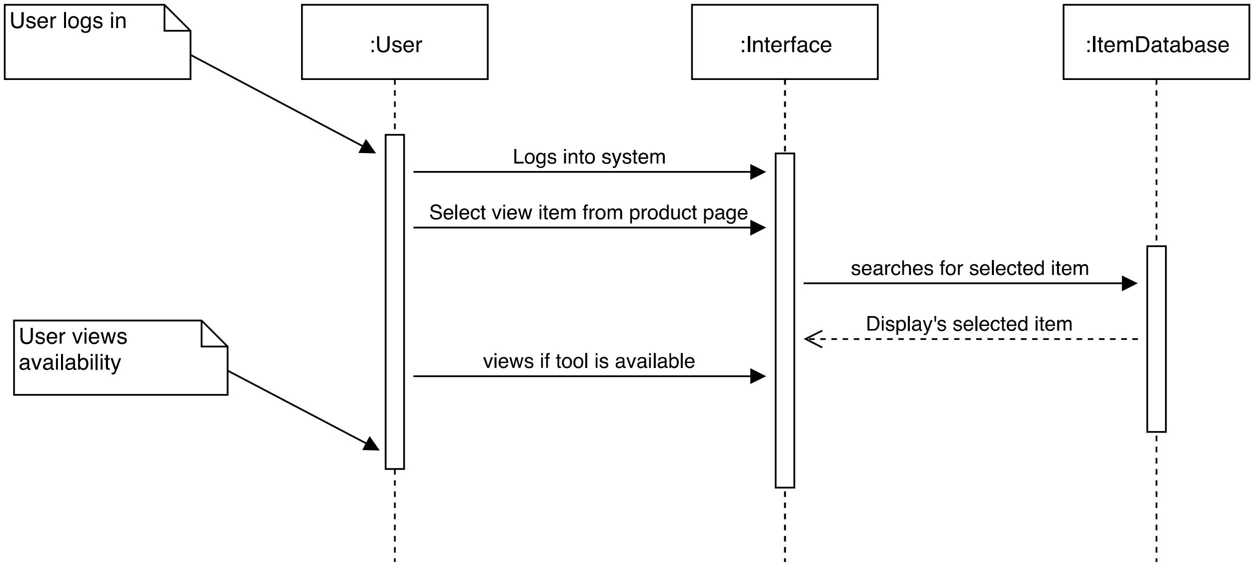
7.2 Login


:User
:Interface
:CustomerDatabase
Info needed is username and password
clicks Log in button
Displays log in page
Enters info
checks if info is correct
false, display error message
true, continue with log in
display successful login
display home page
user successfully logs in

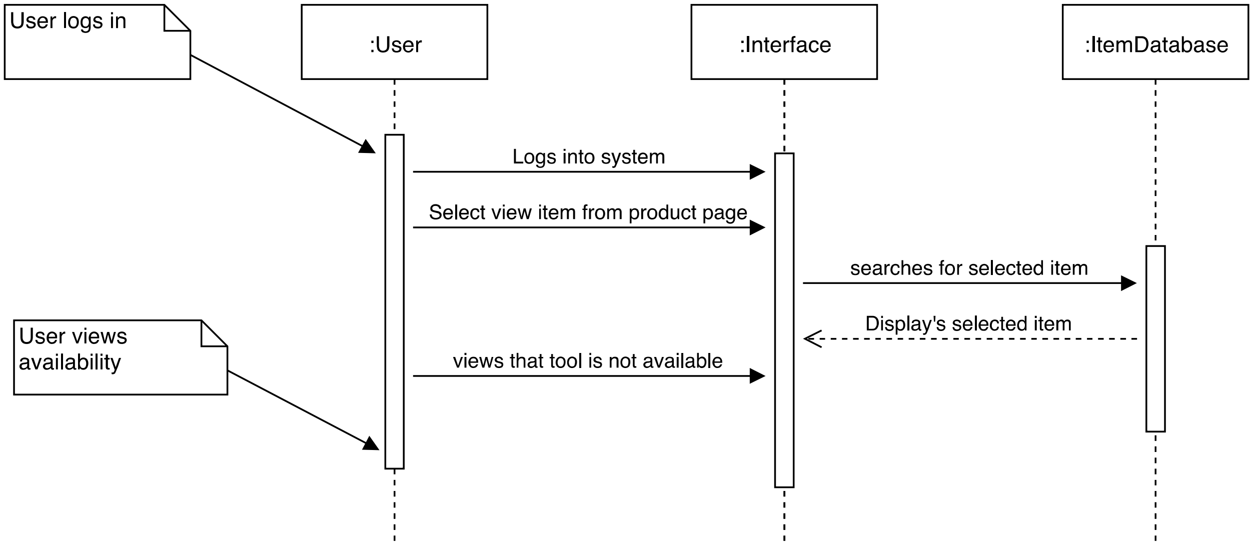
7.3 Search


User logs in
:User
:Interface
:ItemDatabase
Logs into system
Select search from menu bar
searches for an item
Checks if keyword is found
User successfully searched
true, get all results for keyword
false, display no results
Display all results

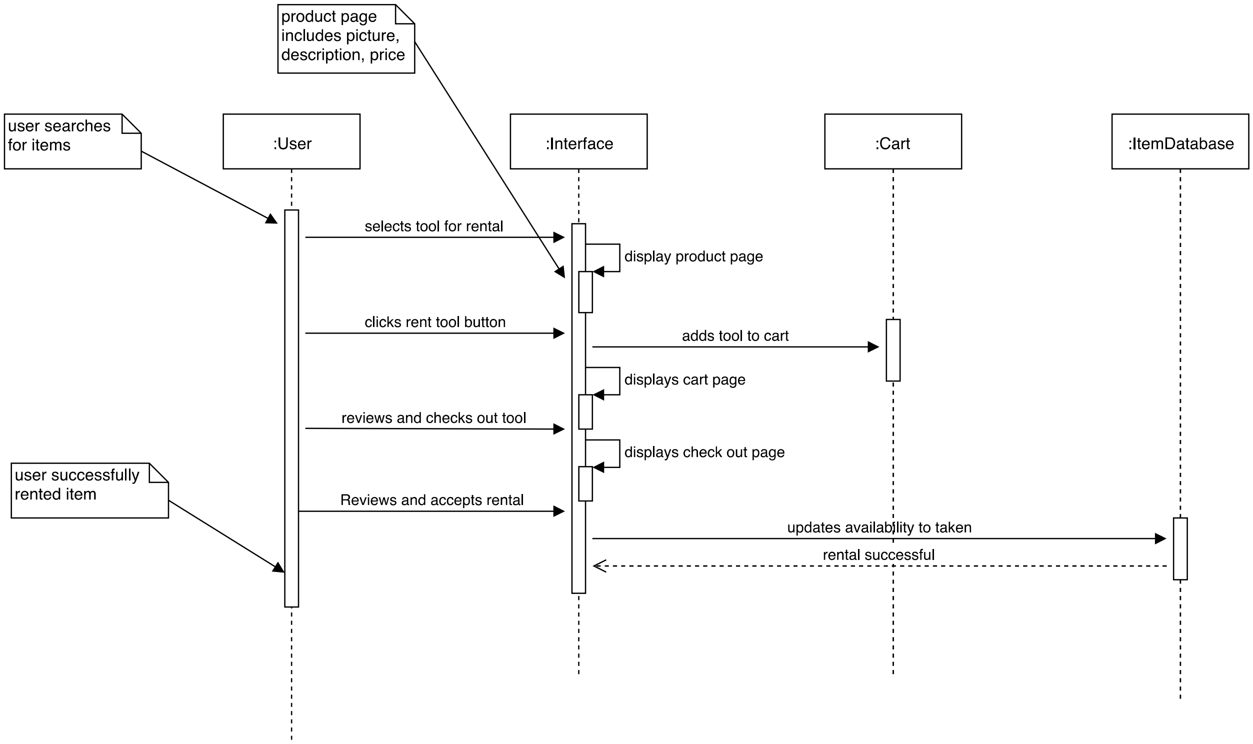
7.4 Check if tool is available



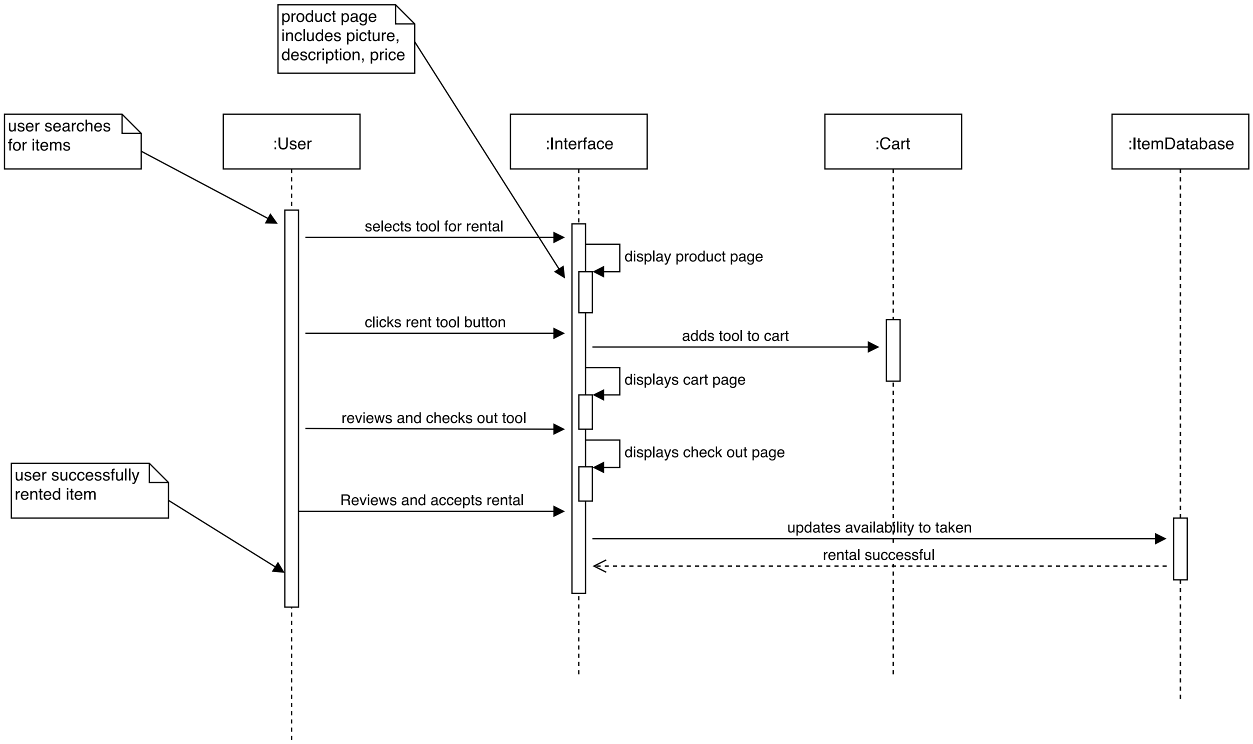
7.5 Check if tool is available (Variation #1)



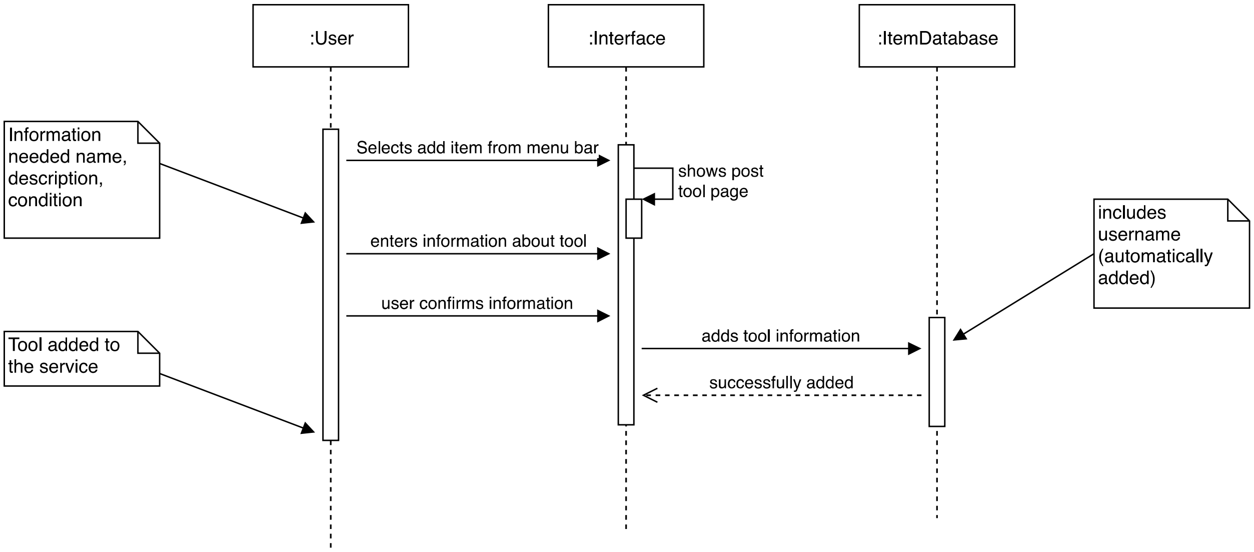
7.6 Rent a tool



7.7 Rent a tool (Variation #1)

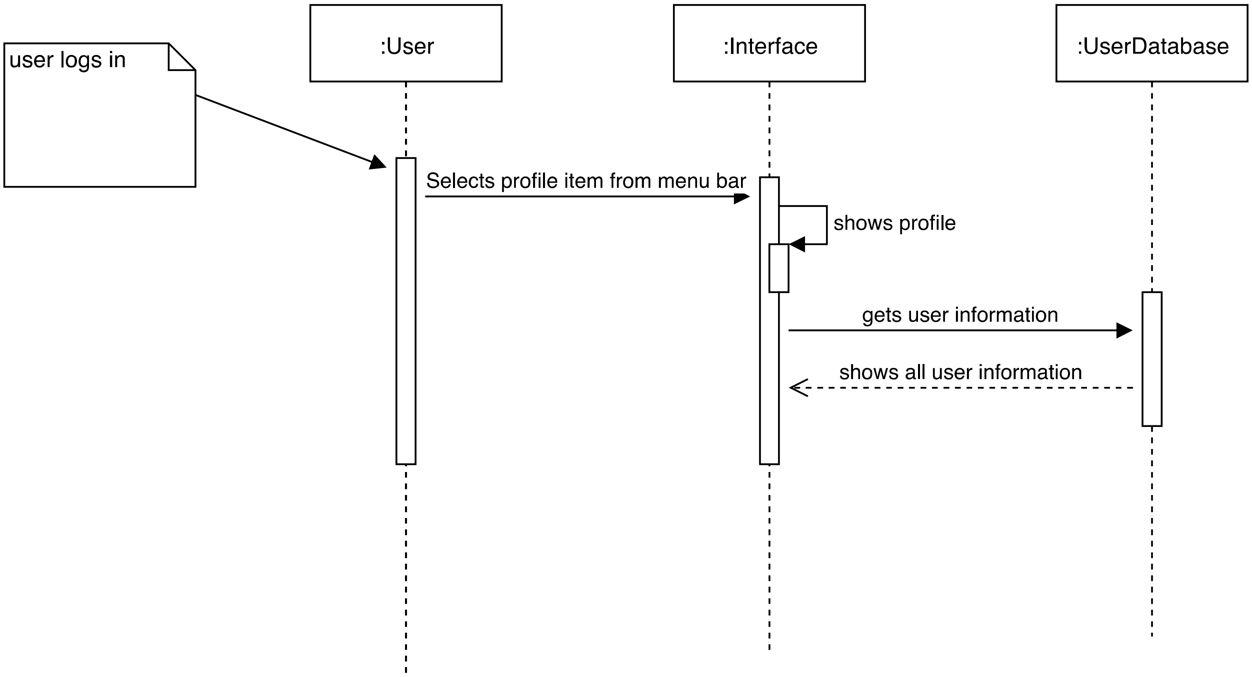


7.8 Post Tool



:User
:Interface
:ItemDatabase
Information needed name, description, condition
Selects add item from menu bar
shows post tool page
includes username (automatically added)
enters information about tool
user confirms information
Tool added to the service
adds tool information
successfully added

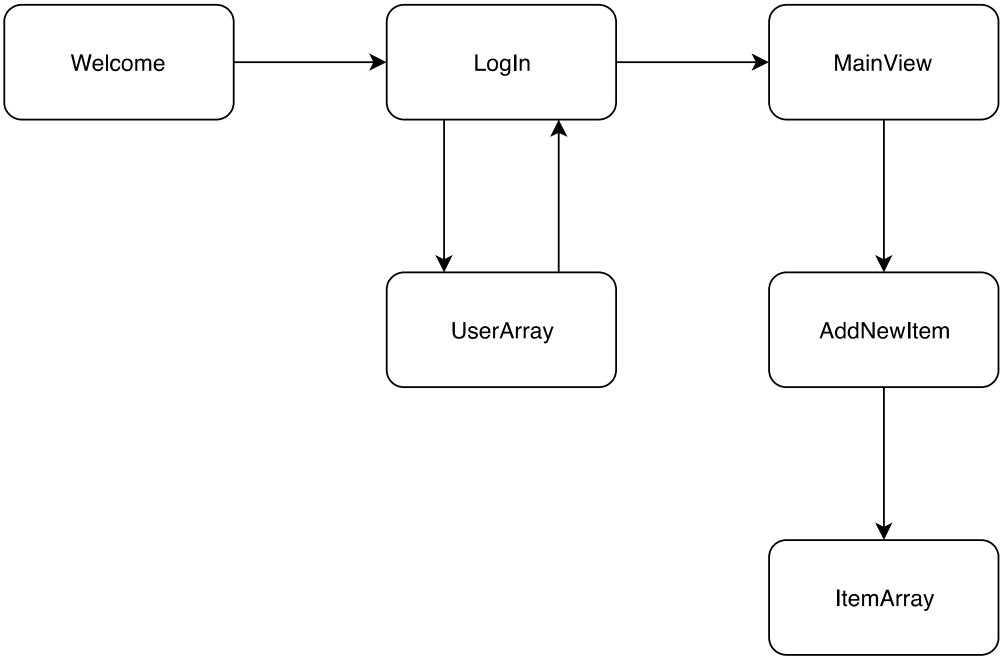
7.9 View Profile



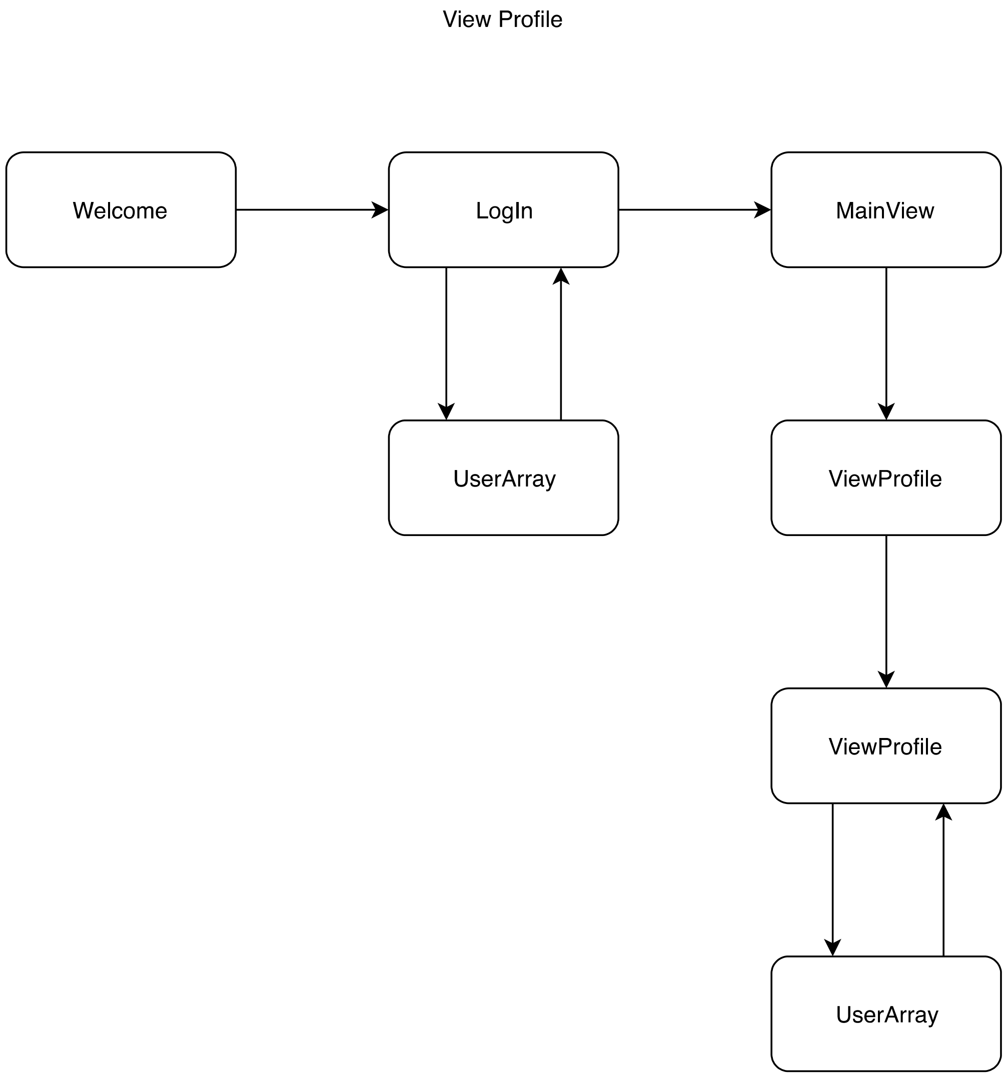
8 State Diagram

8.1 Add new Item

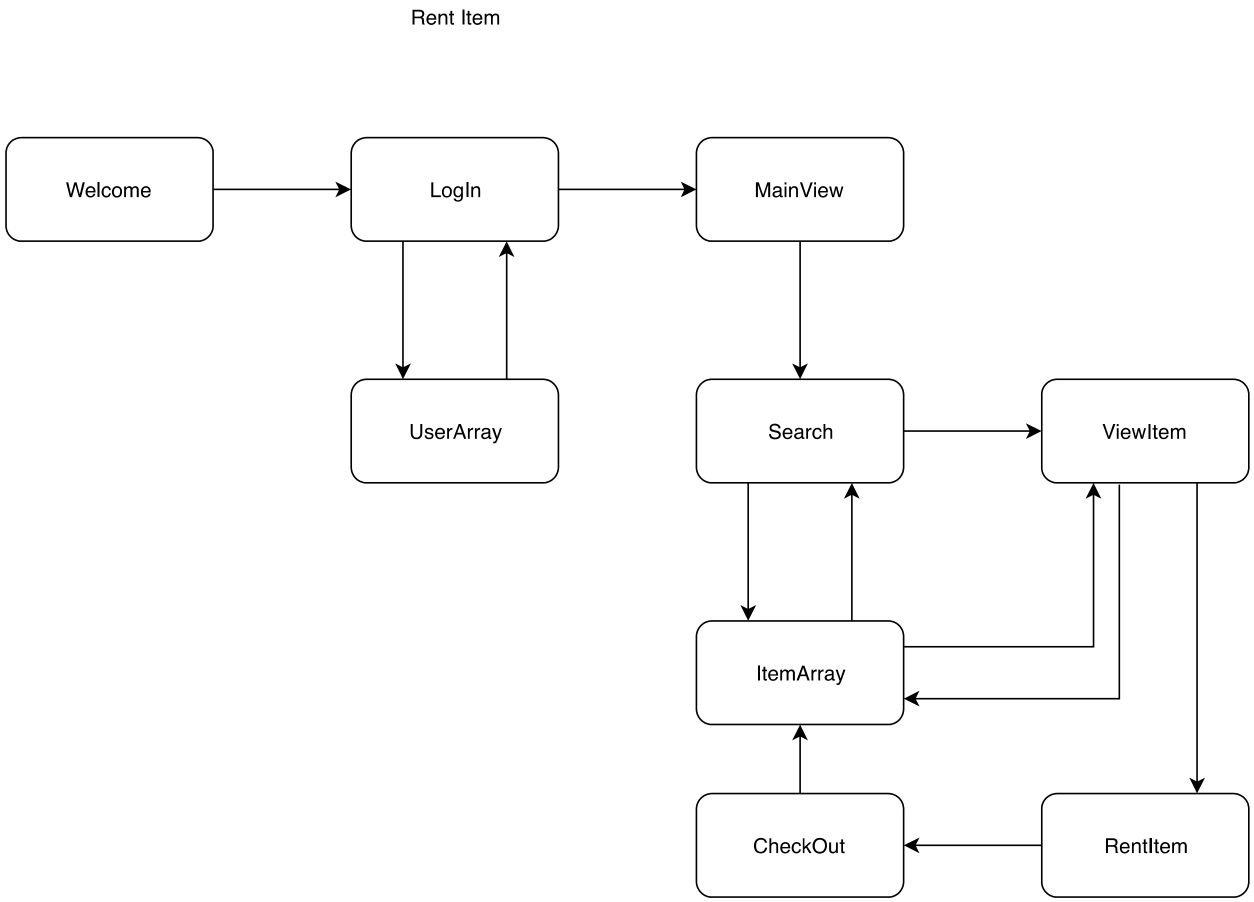
Add New Item



8.2 View Profile



8.3 Rent an Item



9 Design Patterns

1. **<u>Model View Controller (MVC):</u>** Model (Item), View (Product, Search, Cart), Controller (Main Controller for Product page, Login page, etc)

Name in Design Pattern	Actual Name
Model	Item, ItemArray
View	Product, Search, Item, Cart views
Controller	Main Controller for Product Page, login page, etc.

2. **Decorator:** Adding scrollbar to view

Name in Design Pattern	Actual Name
Component	AnchorPane
ConcreteComponent	BorderPane
Decorator	ScrollPane
method()	Initialize()

3. **Iterative:** ArrayList for users, and items and csv files

Name in Design Pattern	Actual Name
Aggregate	List
ConcreteAggregate	ArrayList
Iterator	ListIterator
createIterrator()	Class that implements the ListIterator interface type
hasnext()	hasnext()

4. **Strategy:** How products are seen on screen

Name in Design Pattern	Actual Name
Context	Container
Strategy	LayoutManager
ConcreteStrategy	BorderLayout
doWork()	Initialize()

5. **Composite:** Using a Pane with Label, Button, and ImageView

Name in Design Pattern	Actual Name
Primitive	BorderPane
Composite	Pane
Leaf	Button
method()	Initialize()

10 Lesson Learned

As we mentioned earlier in this report we did learn a great deal through this project. Our tutelage came from multiple sources including class examples, video chats with professor Jarmillo and YouTube videos. We had great ambitions at the outset that quickly changed to realistic goals as we started working.

The first and maybe most important lesson learned was figuring out how to coordinate a project as a team. This was the first major programming assignment that was done as a team for both of us. It is often easier to work on your own while creating a program. Having to coordinate work with another person initially took some adjusting for the both of us. A part of that initial struggle was sharing our code. Your teammate needs to be considered for any adjustment that is made. Another part of the struggle was coordinating design. It is quite typical for two people to consider going about solving a problem in two very different ways. The key for us became communicating as much as possible even if was just a short message. It is important to let your team know what you are working on.

The strength of a team clearly becomes evident as you start working. The amount of work done amazingly doubles. Any road block you hit is not just one person's problem anymore. Any time you are unsure of direction council is readily available. While this team building experience has been a positive one we look forward to the challenges of working in larger teams in our future careers.

A lesson that is evident as a key principle of this course by the very name of the course was reinforced by this assignment. That lesson was that programming separate objects that will then work together in the end makes your program run smoother and life easier in general. It is easier to work on an area of the program that is separate. Make sure it is working properly. Then connect it with the bigger picture after you are sure it functions on its own.

Another lesson that we took away from this assignment was confidence. There is always a way to fix your issues if you look hard enough. Going forward we will have the confidence to take on new challenges with this under our belt.

11 Conclusion

To conclude this assignment, it was not an easy project, where we had to overcome many obstacles. We overcame these obstacles by researching on the internet, discussion it in class with Dr. Jarmillo, asking questions in class, discussing with the project partner, and video chats with Dr. Jarmillo. As mentioned before, we learned a lot during the process of creating the app. It is a great feeling to look back at the progress we made and how everything turned out to a great product at the end. During the development lifecycle, it did not seem that we would finish and make it work in the end, but we succeeded. We did not implement all the features we wanted from our initial plan, which we will try to implement in the future. We adapted our goals along the way as we saw what was and what wasn't possible in the given time frame.

12 Link to GitHub Project

https://github.com/larskoe/ToolDepot_v2