Neighbors

Preston Glenn, Sunita Aryal, Connie Bardalez, Ritesh Manandhar,Akshay Durvasula, Ram Panda, Advait

Objective

A rental website where users can request to rent items from people who live near them for a short period of time. For example, if someone needs a high-quality camera or speaker system for a day but does not want to own it permanently, they can pay someone in their neighborhood (who has the items) a small fee to borrow the items and then return them.

Motive

We know that in hard economic times, it is very difficult to buy a lot of expensive luxuries at full price. With many people living paycheck-to-paycheck, financing big purchases may not work out or even worse, may end up costing more than the original item in the long run. Therefore, why not borrow your neighbor's high-quality speaker system for a party, for example, and then give it back a couple of days later for a small fee instead of paying outrageous prices for it.

Even if one is not under economic strain, there are many things that we can enjoy for short periods of time so that we don't have to have those items sitting in our house collecting dust. Vice versa, if someone happens to have a big-time purchase item such as a flatbed trailer or floodlight that they would like to make money from, Neighbors.com can be their go-to location to rent out their item.

Requirements

Functional Requirements

- User login and profile creation
- Browsable and filterable rental listings
- Transactions approval functionality.
- Price negotiation and bargaining capability.
- Messaging and communication functionality between renters and lenders

Non-Functional Product Requirements

- **Efficiency**
  - App should not take more than 4 seconds to load items and start up
  - Application size should not exceed 200 MB
  - Performance of the application shouldn’t change drastically if more users are using it at the same time
- **Usability**
  - Users should be able to locate features and navigate throughout the app with ease and app layout should be user friendly
- **Dependability**
  - Users should receive confirmation via email or text after each purchase
  - System shouldn’t experience many crashes especially during the payment process
- **Security**
  - Personal and payment information of every user shouldn’t be accessible to unauthorized personnel
  - Users should not be able to modify any critical data on the app.

Non-Functional Organizational Requirements

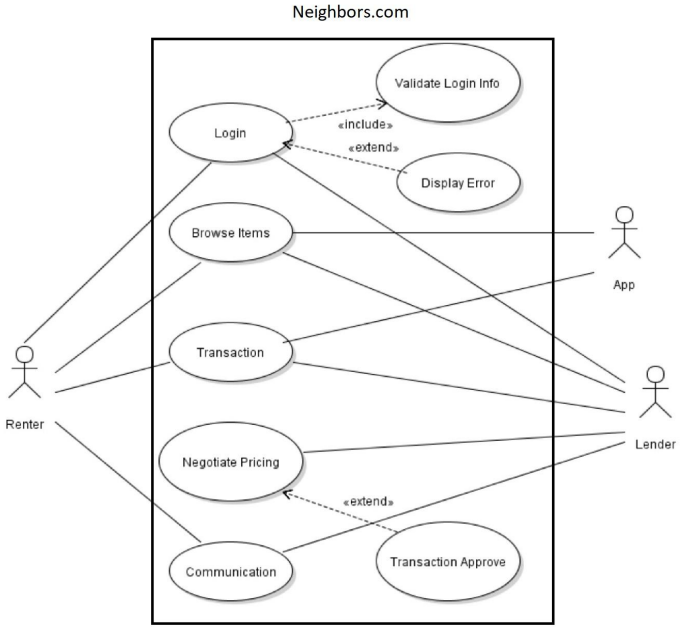
- **Operational**
  - Must be compatible with iOS and android devices.
  - App must be scalable to accommodate any large growth of users
- **Environmental**
  - App must be able to adapt to different environments and configurations
- **Development**
  - Any future modifications or updates should be simple and cost efficient

Non-Functional External Requirements

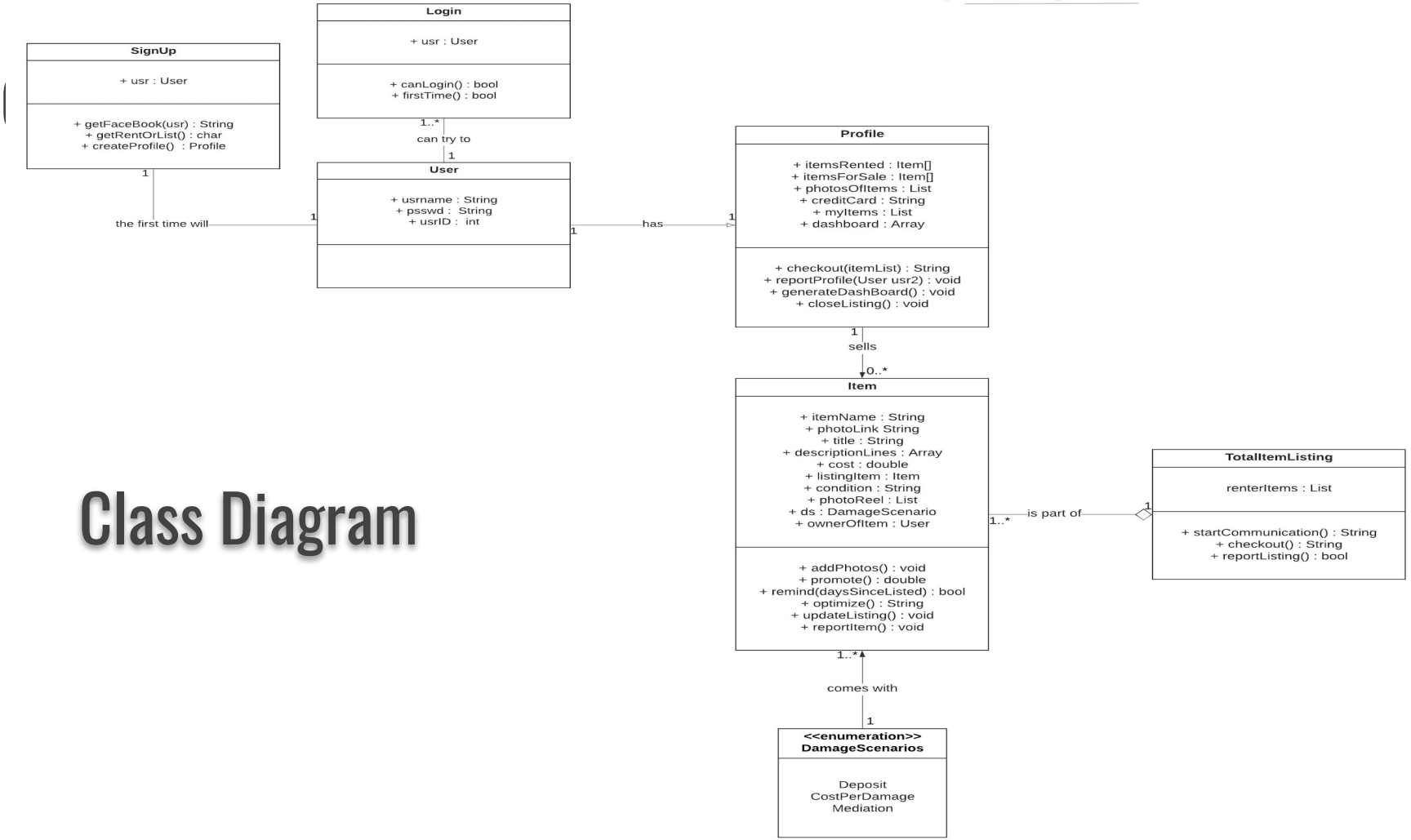
- **Legislative**
  - Stolen or illegal items cannot be listed in the application
  - Users should not post messages that violate rights or privacy of other users
  - Harvesting user info is prohibited
- **Ethical**
  - Users should not list inaccurate prices or prices which they don't intend to sell
  - Users shouldn't harass or threaten other users through messages
- **Regulatory**
  - Prohibited items such as drugs, alcohol, weapons, or adult content should not be listed

Diagrams

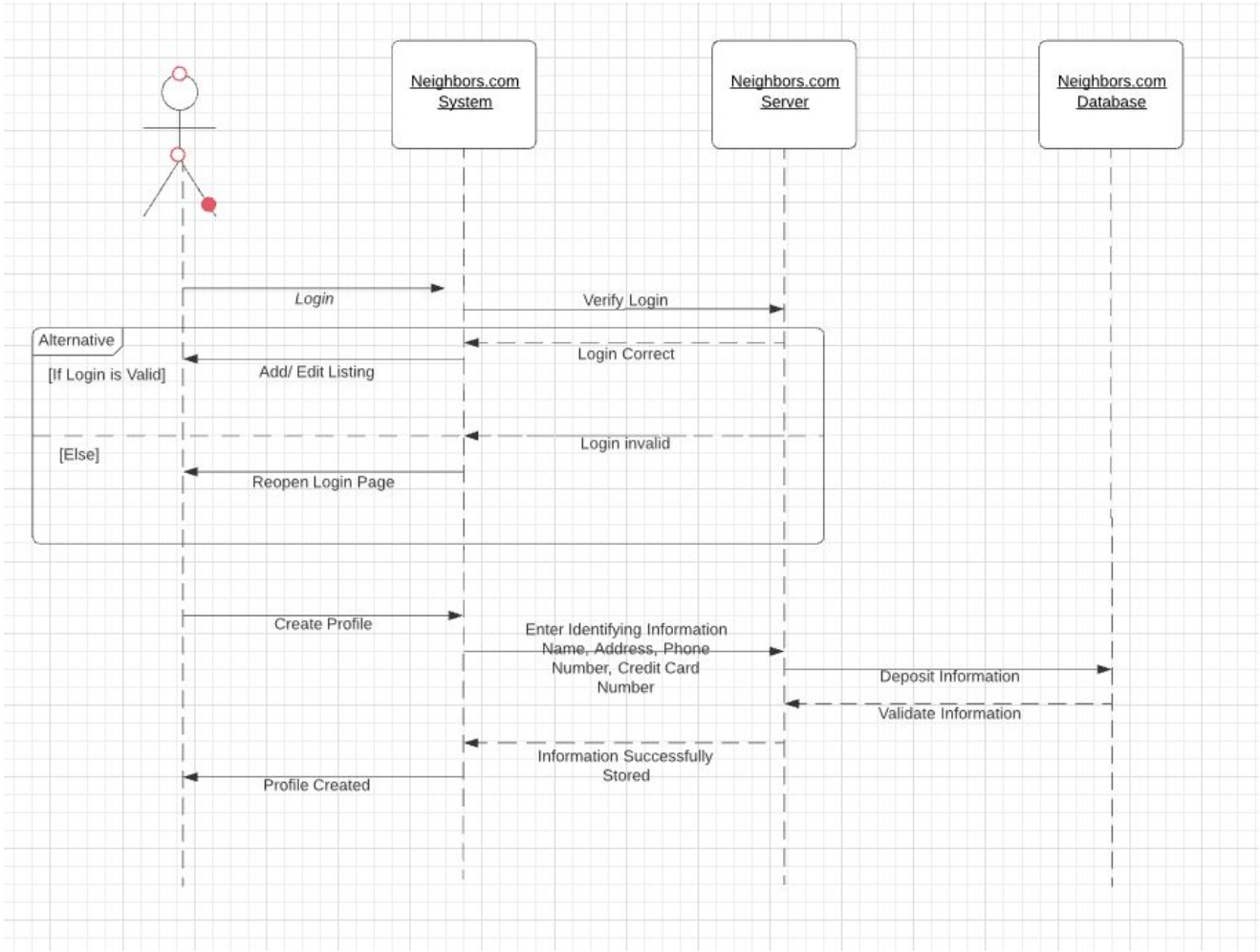
Use Case Diagrams



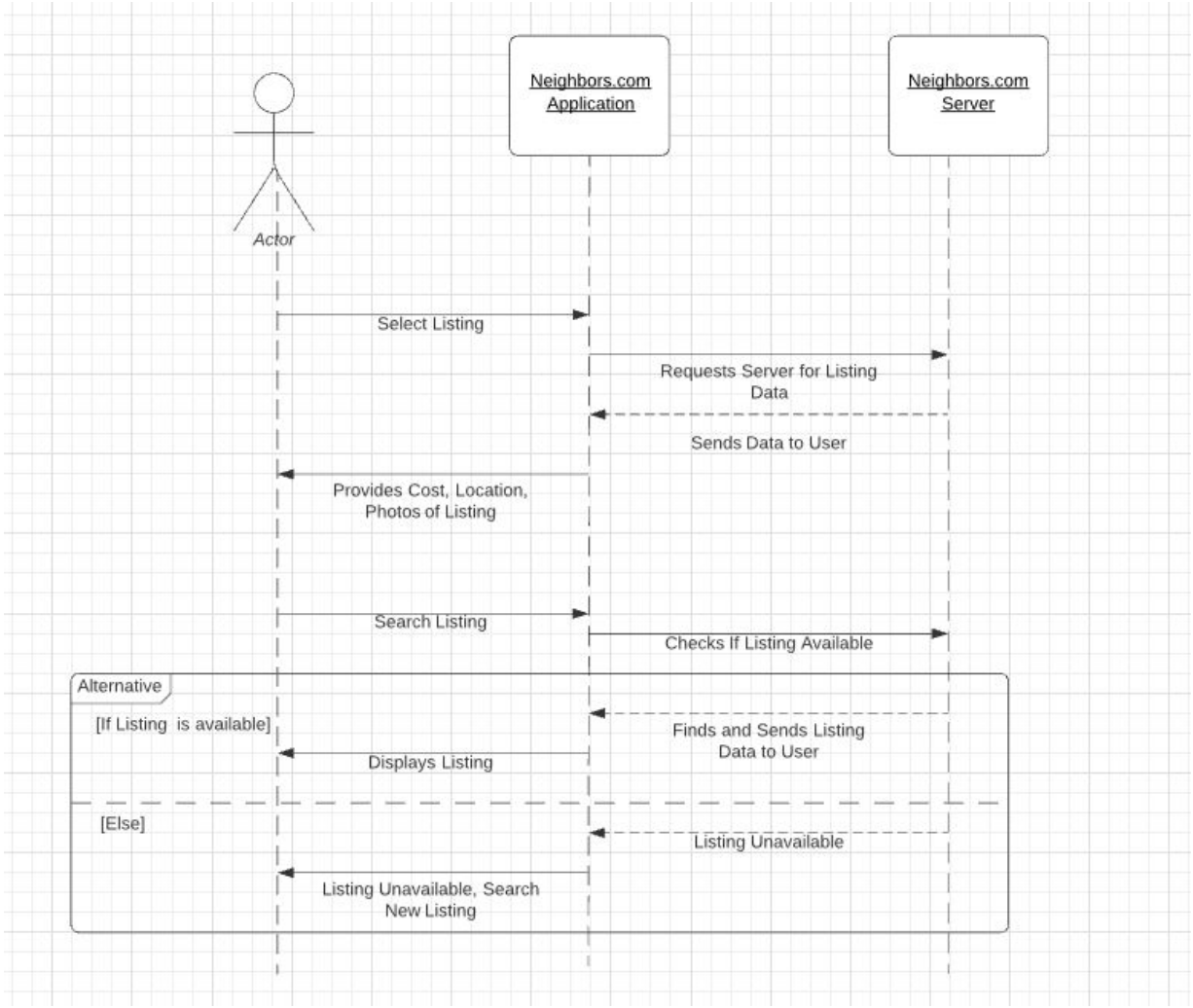
Class Diagram



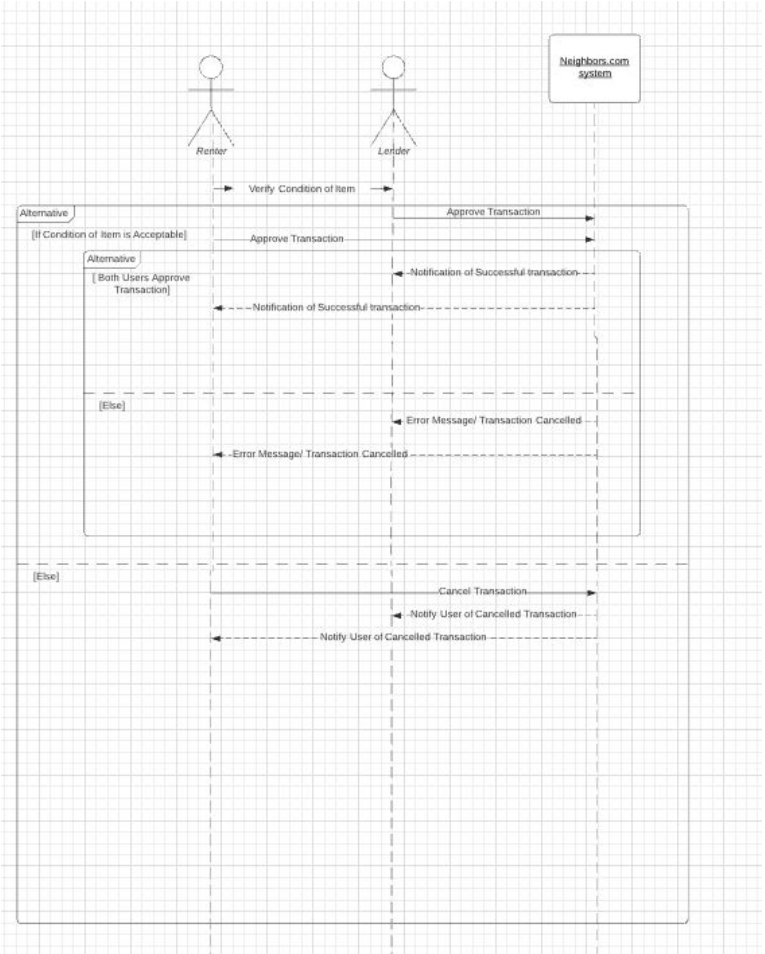
Sequence Diagrams-Login



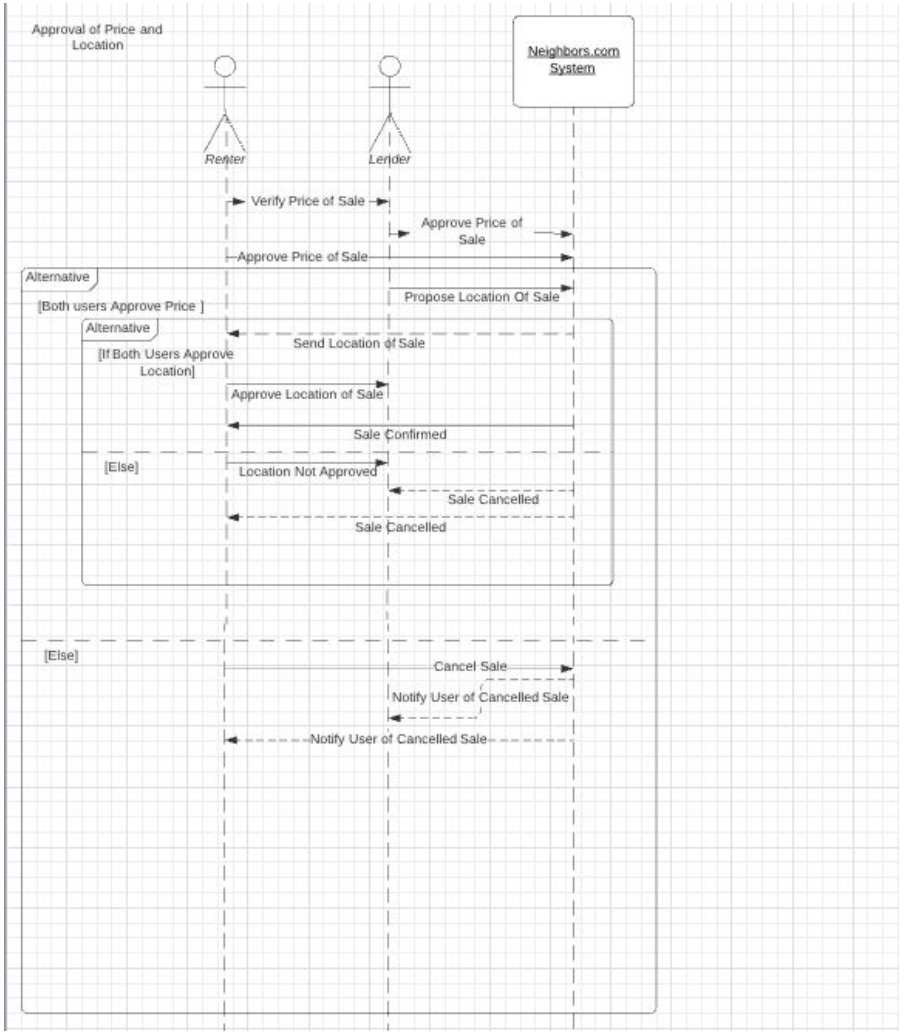
Sequence Diagrams - Browse Listing



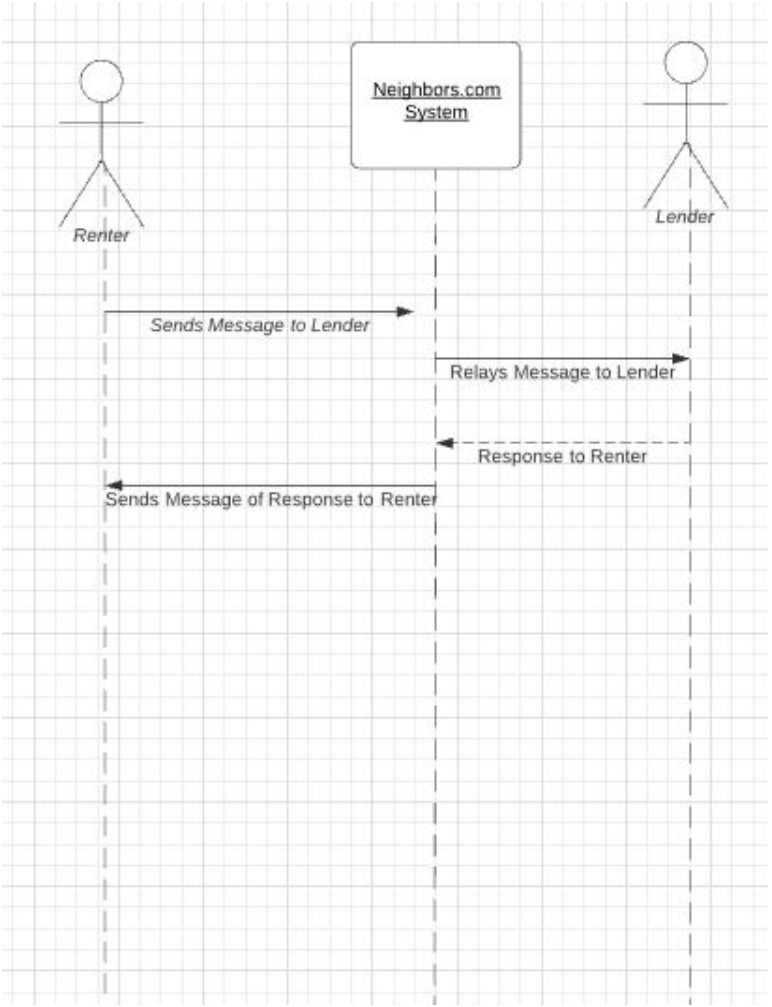
Sequence Diagrams - Transaction



Sequence Diagrams- Approval of Price, Location



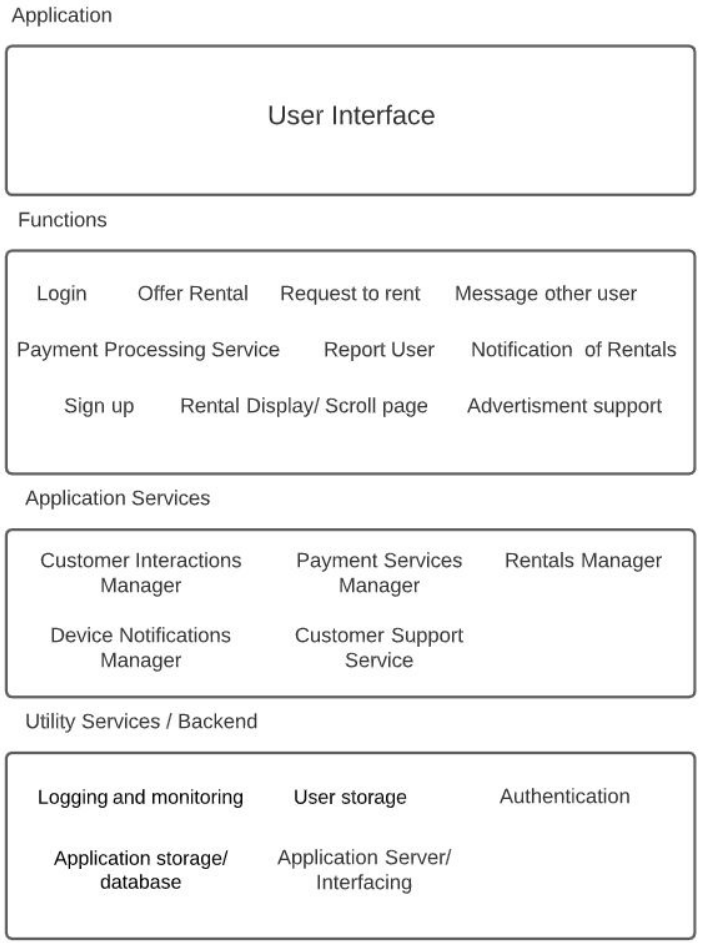
Sequence Diagrams- Communication



Architecture

Layered Architecture Approach

- Top down approach: services support layers above
- Easily Refactorable
- Easily compartmentalizable



Estimations

Project Timeline Estimation

Task estimation

Task	Description	Start Date	Due Date	Department	Assigned to
Plan Front-End functionality and Requrements	Design the Front-end functionality and the Languages and Capabilities neede	10/12/20	12/07/20	Technology	Akshat
Backend	Choose Data hosting service based on developers' skill levels and budget	12/07/20	12/21/20	Technology	Connie
Authentication	Create client to database authentication by using a token of choice	12/21/20	01/10/21	Technology	Sunita
Budgeting	Compile the results of budget calculations from other planning modules	01/10/21	01/18/21	Accounting	Ram
Create Logo and color scheme for Interface	Develop a logo and colors for the website	10/27/20	11/24/20	Design	Ritesh
Create Profile page feature	Set different planned features for lenders and renters	01/18/21	03/01/21	Technology	Advait
Add Login Page	Different login page for different login methods	01/18/21	03/01/21	Technology	Preston
Security Configuration	Study how to set up server authentication	03/01/21	03/15/21	Technology	Akshat
Maps Functionality Development	Develop Google maps integration	03/15/21	04/12/21	Technology	Connie
Payment Services	Explore options for payments and budget for licensing to different payment au	03/15/21	04/12/21	Accounting	Sunita
Add functionality to post Items	Create functionality to be able to post items for rental	04/12/21	05/10/21	Technology	Ram
Add ability to rent items	Create functionality to be able to pay and schedule items for rental	05/10/21	06/07/21	Technology	Ritesh
Develop Messaging Functionality	Messaging functionality between renters and lenders	05/10/21	06/21/21	Technology	Advait
Add search and filter functionality	Search and filter for items by distance, category	05/17/21	06/21/21	Technology	Preston
Hosting Services	Choose a hosting service for the number of users and budget depending on sp	05/24/21	05/31/21	Technology	Akshat
Add damage reporting and complaints feature		05/31/21	06/07/21	Legal	Connie
Integration Testing	Finish all full unit testing and integration testing	06/21/21	07/05/21	Technology	Sunita
Take out commision from rental fees	Later on, we take a part of the rental fee as revenue	06/07/21	06/21/21	Marketing	Ram
Implement feedback and bug reporting services	Create text boxes to submit complaints	05/31/21	06/14/21	Legal	Ritesh
Create mobile app	Lead team of developers to create the mobile app	11/24/20	02/16/21	Technology	Advait

Hardware Cost Estimation

	Description	Price	Quantity	Total
Laptops	i5 14in 8GB mem 256GB SSD	$899	15	$14,597.51
Monitors	24-inch FHD	$109	5	$589.96
Mouse	2.4 wireless mouse	$8.66	15	$140.62
Router	Dual Band Gigabit Wireless router	$59.99	1	$64.94
				$15,393.03

Software Cost Estimation

Authentication Service 50,000 users	Phone authentication - 400/month Since our project will be 32 weeks long therefore 8 months *400 = $3,200 total
Intellij IDE Subscription	$499 *15 per user, yearly = $7,485
Realtime Database	$10 per month [2] $10 x 8 months = $80
Storage	$33.58 per month [2] $33.58 x 8 months = $268.64
Cloud Functions	$41.06 per month [2] $41.06 x 8 months = $328.48
iOS Development	a simple iOS app like ours with basic functionality thats takes up almost about two months to build will cost about $30,000
Total	$41,362.12

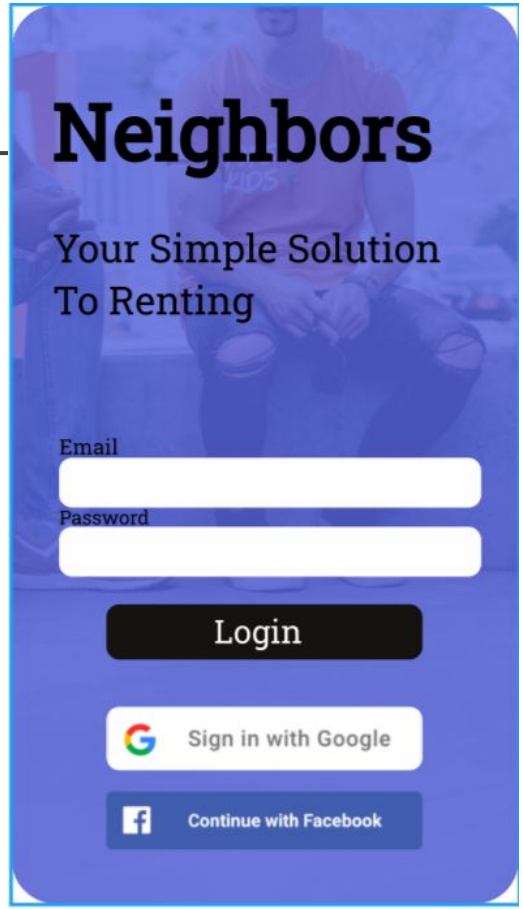
Estimated Cost of Personnel

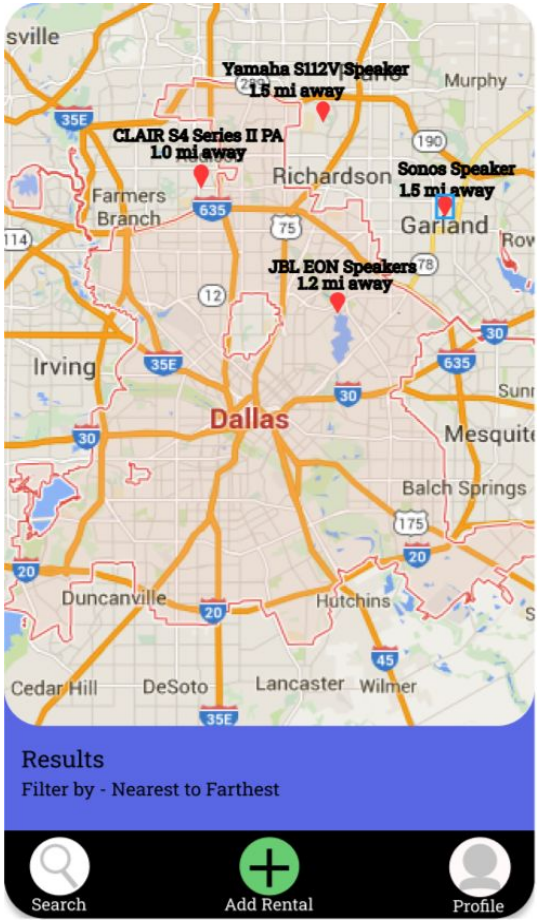
Software product cost estimate:	**Roles**	**Subscriptions**	**Cost**
Marketing and Product management	Provides app with updates and customer support.	$35 hourly 1 person	32 weeks project 32 weeks * 40 hours/week = 1280 hours 1280 * 35- * **$44800**
Unit and integration testing	Engineers monitor every phase of the software development process. QA managers make sure that new products are bug-free.	$25 hourly 4 people	4 weeks project 4 weeks * 40 hours/week = 160 hours 160 hours * $25 * 4 = **$16000**
UI/UX designer	person on which depends on the app structure and visual appearance.	$25 hourly 2 people	32 weeks project 32 weeks * 40 hours/week = 1280 hours 1280 * $25 * 2 = **$64000**
Backend and database maintenance including storage	creates the app back-end infrastructure and API integrations.	$ 30 hourly 2 people	10 weeks project 10 weeks * 40 hours/week = 400 hours 400 hours * $30 * 2 = **$24000**
Developers	Range from $50 to $250 per hour	$50 hourly 6 people	32 weeks project 32 weeks * 40 hours/week = 1280 hours 1280 * $50 * 6 = $**384000**
Miscellaneous Cost			**$2,000**
Total		15 people total	**$543800**

Final Estimated Cost

Hardware Products + Software Products + Cost Personnel	Total Estimated Cost
Estimated Cost	$600,555.15

Project Demo





Comparison to Existing Products

- The closest real-world software comparison to our idea would be the highly popular websites, Facebook Marketplace and eBay.
- Both platforms allow users to buy and sell virtually any item they own. Since the marketplace's social media counterpart, Facebook, already creates a full profile for users, the marketplace uses the same profiles for its website.
- Neighbors.com introduces a key change to the framework of the facebook marketplace and eBay. Namely, customers do not *buy* an item, they *rent* it. This creates another layer of sophistication that our system has to manage for returning items back to users.
- Neighbors.com only features items from lenders that are exclusively within a certain radius of the user's location while facebook marketplace and eBay also show items from out of the set radius.(as in, items that are sold outside a specified radius can still be purchased).

Conclusion

- Changes in design made throughout the project
  - For example, when we knew that we needed users to be able to communicate with each other in order to exchange the item, however, at first we forgot to consider the fact of how customers would actually approve the transaction and indicate that they had received/given out the item that was being rented.
- The changes allowed our concept to become more stable