
# Home Sweet Home Business System

By LLMS

Group 11

Joshua Armstrong

Benya Chongolnee

Mathew Berry

Joey Mucha

Kyle Kmetz

## Executive Summary

Home Sweet Home Property Management System currently is very outdated because all of the documents are saved and stored using a pen-paper system. Nothing is online and the system has a high chance of some documents being lost or destroyed. There currently is no security for Home Sweet Home's management system. The Home Sweet Home Business System by LLMS is the solution that can help Home Sweet Home organize their system and help the company be more technologically advanced.

The purpose of this document is to show the in depth steps that were taken in order to ensure a successful setup of the Home Sweet Home Business System by LLMS. The following document includes a product vision statement, a context diagram, a system request, a use-case diagram, use-case narratives, activity diagrams, a class diagram, sequence diagrams, an entity relationship diagram, a deployment diagram, the system's screenshots, cash flow, a gantt chart, necessary details regarding the system and a preliminary usability test.

The Home Sweet Home Business System by LLMS is a software that will be used by the staff, maintenance, buyer, and seller. This project should take no longer than 3 months and should cost less than $120,000 with the benefit of more than $150,000.  This system will be able help each user as well as the whole company be more organized and effective; therefore, resulting in better and more profitable business.

# Table of Contents

# Revised Milestone Content

## Introduction:

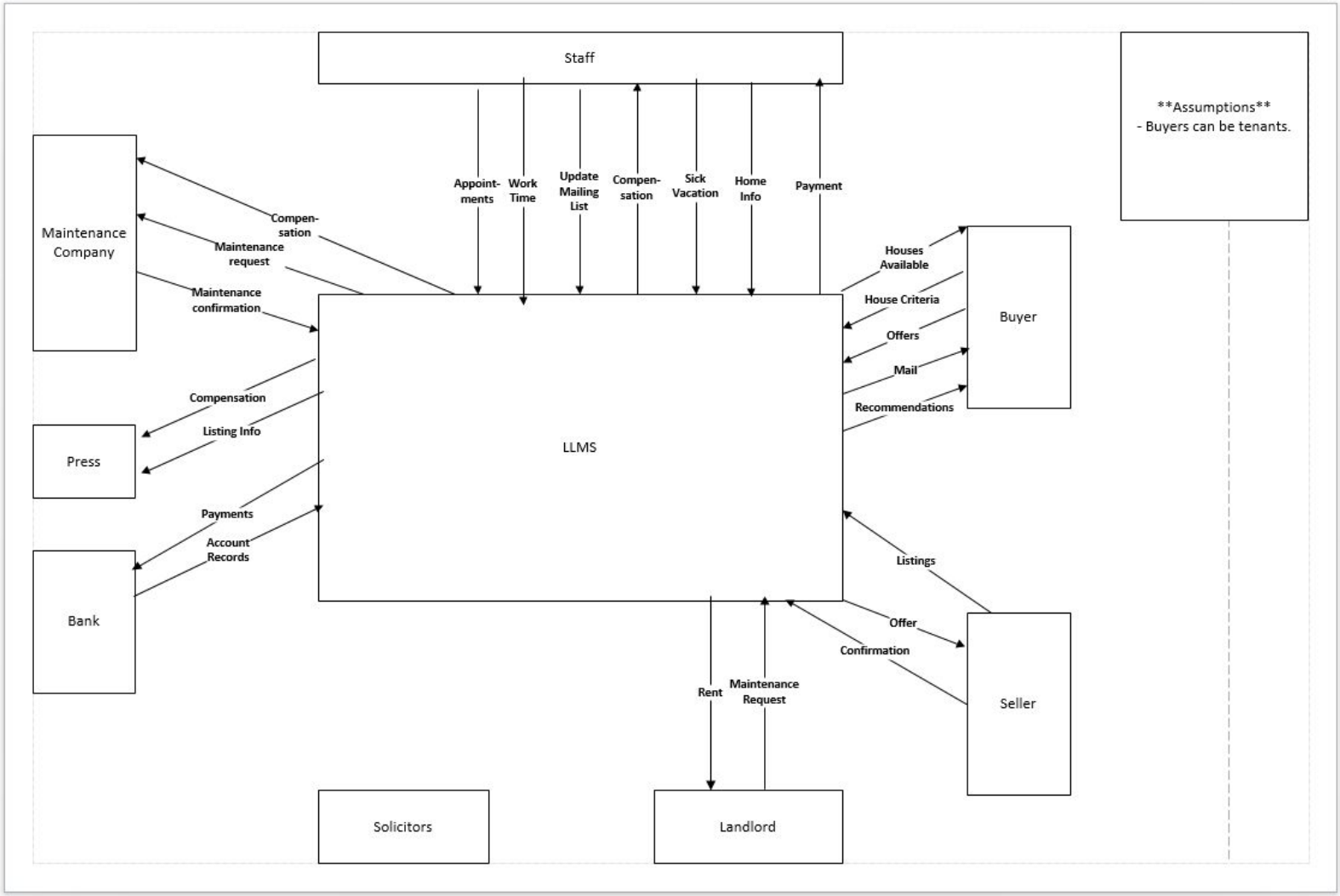
The purpose of this document is to show what we have done so far as LLMS to build a system for Home Sweet Home. We want to be on track when making this system so that we can finish this project in time and give it quality work. We want to get a sense of where we are right now during this project to see how much there is left to do. This report will mention the vision statement, system request, product backlog, future plans, and context & use-case diagrams. With that being said, we hope that this report is able to give the LLMS better sense of the work there are left to do and shows the Home Sweet Home's staff our process thus far.

We are the LLMS and we build software solutions for businesses that need better Information Technology infrastructure. We are currently building a system for Home Sweet Home in order for their business to run smoothly and efficiently. We will be implementing the Home Sweet Home system using the Business Process Reengineering method. We chose this method because our proposed system is completely different from the current system. We needed to total transform the whole system using the BPR system. This system's potential business value is very high; however, the project cost is high also.

## Project Vision Statement:

For Home Sweet Home staff and customers who need a better business system, Home Sweet Home Business Systems, is a real estate management system that can virtually and easily manage all aspects of Home Sweet Home from accepting rent to scheduling appointments. Unlike traditional pen and paper systems, our product is easier to use and more convenient.

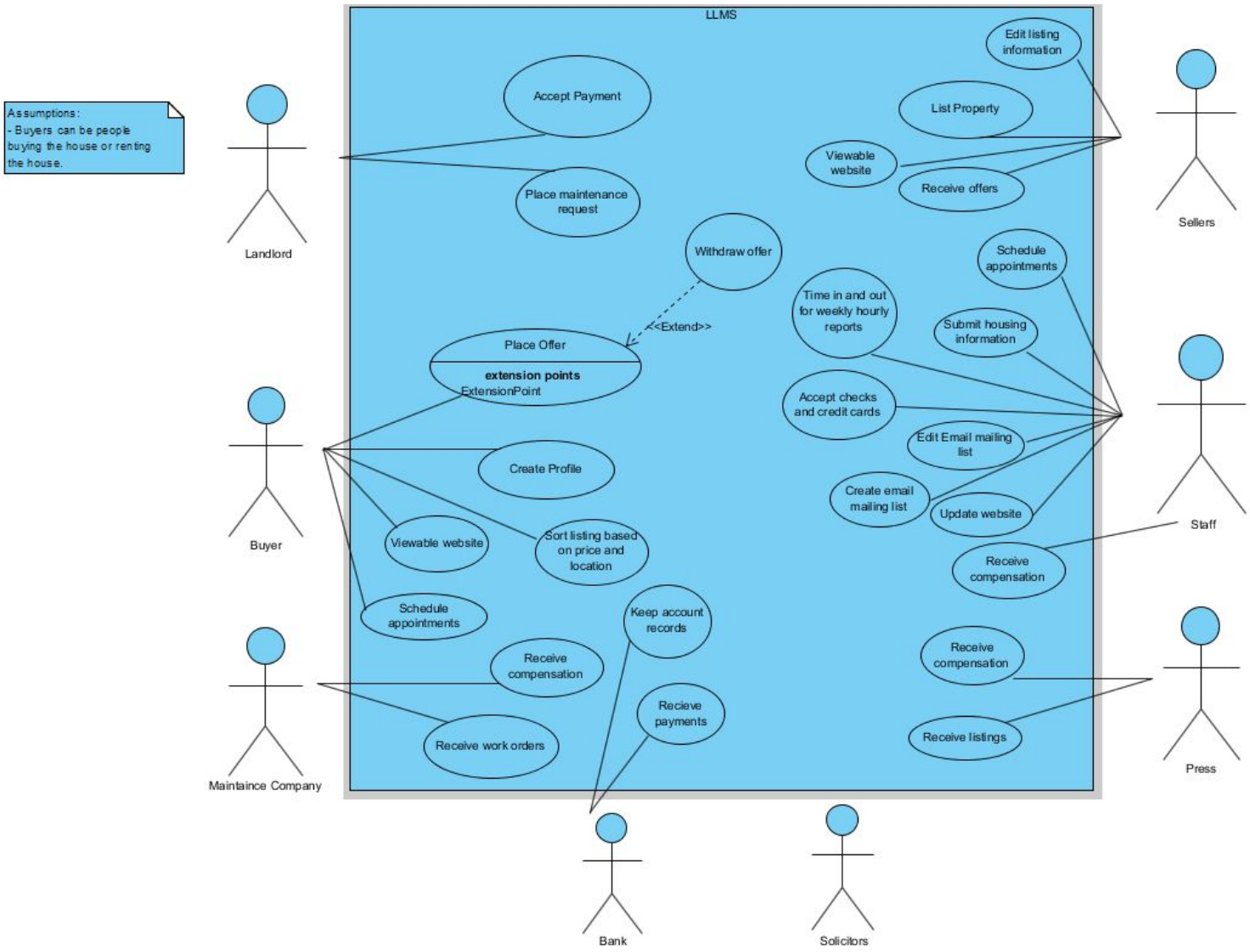
## Context Diagram:



**System Request:**

The project sponsor for this system is the Home Sweet Home staff because the staff expressed a need for a new and technologically-advanced system. The staff of Home Sweet Home will be the contact person when problem arises and/or if needed more necessary information. They will be the first group of people to see the product and give opinions regarding the system. Home Sweet Home does every business processes by hand, and keep physical records of everything. This increase the chance of wrong information being recorded, increase time in finding records, and increases the possibility of lost of records. This is why Home Sweet Home need the help of LLMS. We are able to help create a new business system that will be efficient and easier to keep track of. Everything the staff and customers of HSH recorded, will be kept in the system. The requirements of the system is for the staff to easily change information, buyer to buy and look up listings, seller to sell their houses and receive offers, landlords to submit payments, maintenance to see what jobs are available, and for the press to receive updated listings. The tangible benefits of this system will be that it is faster, and more efficient in achieving the results demanded by the staff and customers who use it. The intangible of this system is happier staff; therefore, higher productivity. This then will lead to higher customers satisfaction and overall increase in customer base. Special issues for Home Sweet Home would include a 25,000 pound cost constraint.

## Use Case Diagram:



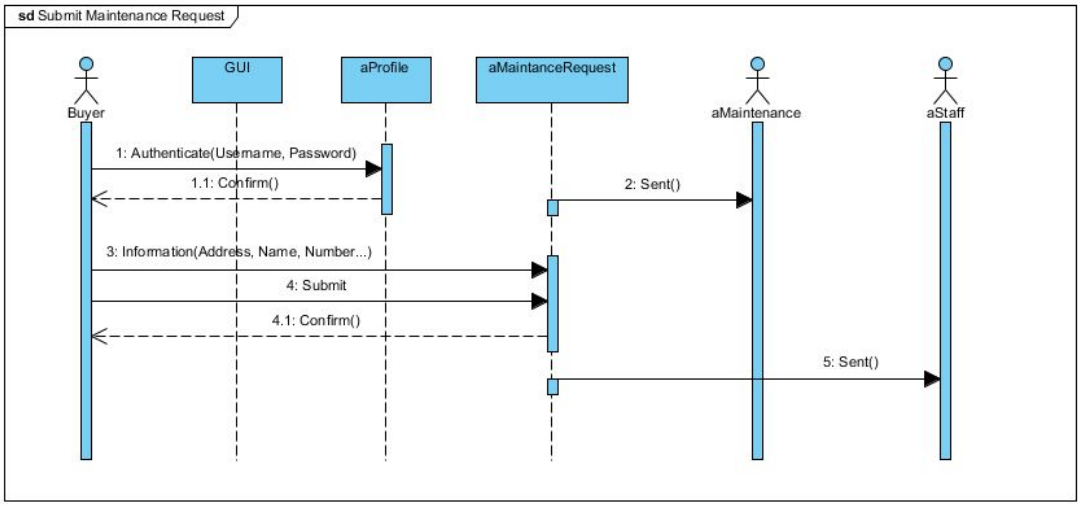
**Use case Narratives & corresponding diagrams:**

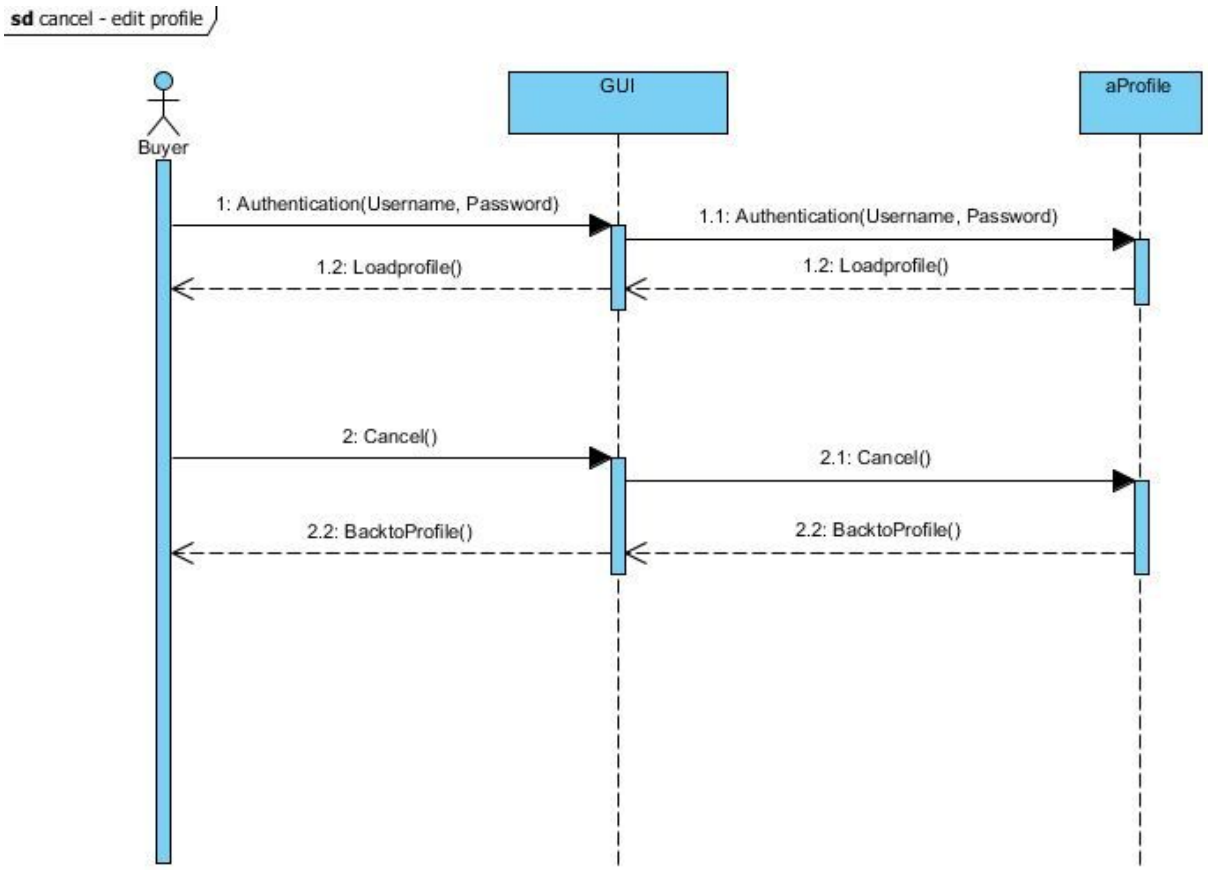
*(please note that the author of the 10 narratives we are providing also did their corresponding diagrams)*

Author: Benya Chongolnee Date:10/12/16

| | | |
|---|---|---|
| Use-Case Name: | Edit account profile | Use Case Type<br>Business Requirements: X<br>System Analysis:<br>o<br>System Design:<br>o |
| Use-Case ID: | 003 | |
| Priority: | Mid | |
| Source: | Backlog | |
| primary business actor: | Buyer | |
| Primary System Actor: | Buyer | |
| Other Participating Actors: | Seller and staff | |
| Other Interested Stockholders: | None | |
| Description: | Buyers are able to edit their account profiles such as name, current address, their budget, and what they want in a house. | |
| Precondition: | Buyer already has an account | |
| Trigger: | The seller clicks on “edit profile” button on their profile page | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1) Clicks on “edit account” button on their profile page<br>3) Users change the profile | 2) System generates their current profile information |

| | | |
|---|---|---|
| | information they want to edit<br>4) Users clicks save | 5) System saves the information<br>6) Send confirmation to the user |
| Alternate Courses: | 4a) If the user did not change any information, they can click cancel. | |
| Conclusion: | New information for the buyer are updated | |
| Postcondition: | Return user to their profile page | |
| Business Rules: | none | |
| Impl. Constraints and Specifications: | none | |
| Assumptions: | Buyer already has an account | |
| Open Issues: | none | |

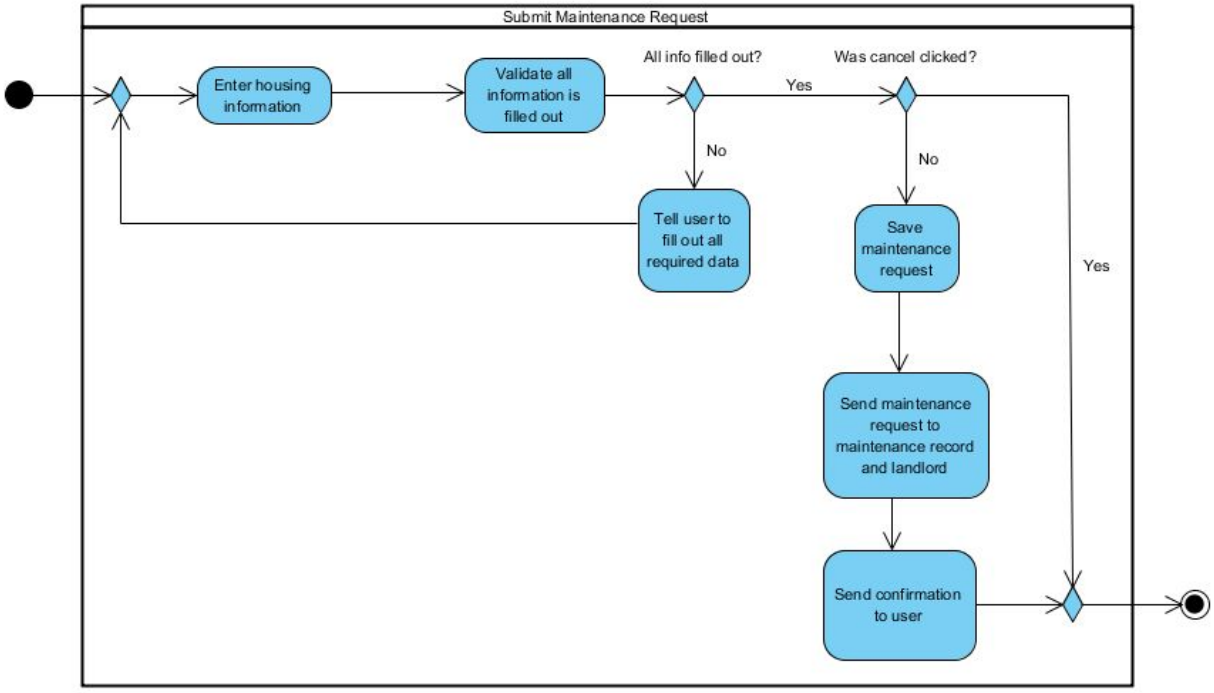

sd Submit Maintenance Request
Buyer
GUI
aProfile
aMaintanceRequest
aMaintenance
aStaff
1: Authenticate(Username, Password)
1.1: Confirm()
2: Sent()
3: Information(Address, Name, Number...)
4: Submit
4.1: Confirm()
5: Sent()

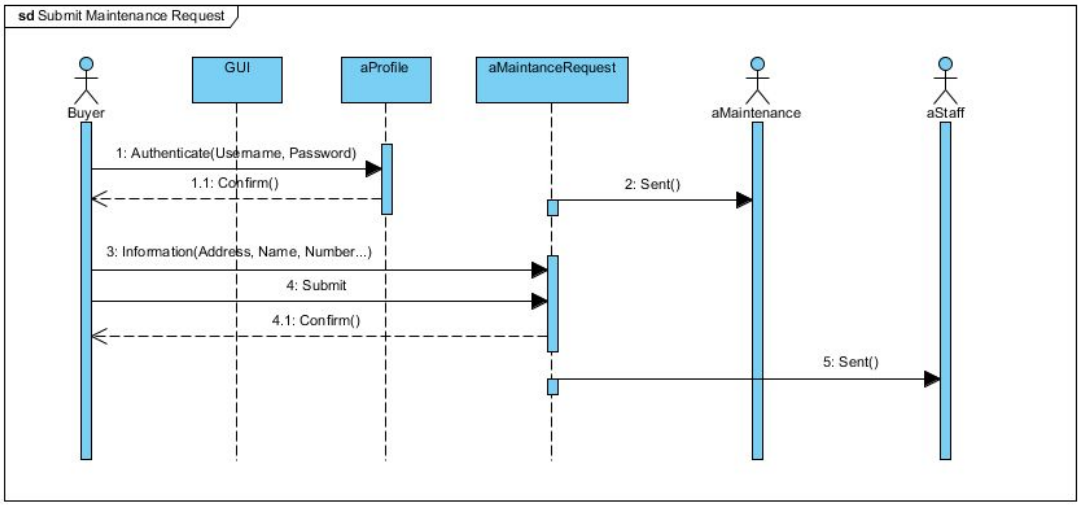

sd cancel - edit profile
Buyer
GUI
aProfile
1: Authentication(Username, Password)
1.1: Authentication(Username, Password)
1.2: Loadprofile()
1.2: Loadprofile()
2: Cancel()
2.1: Cancel()
2.2: BacktoProfile()
2.2: BacktoProfile()

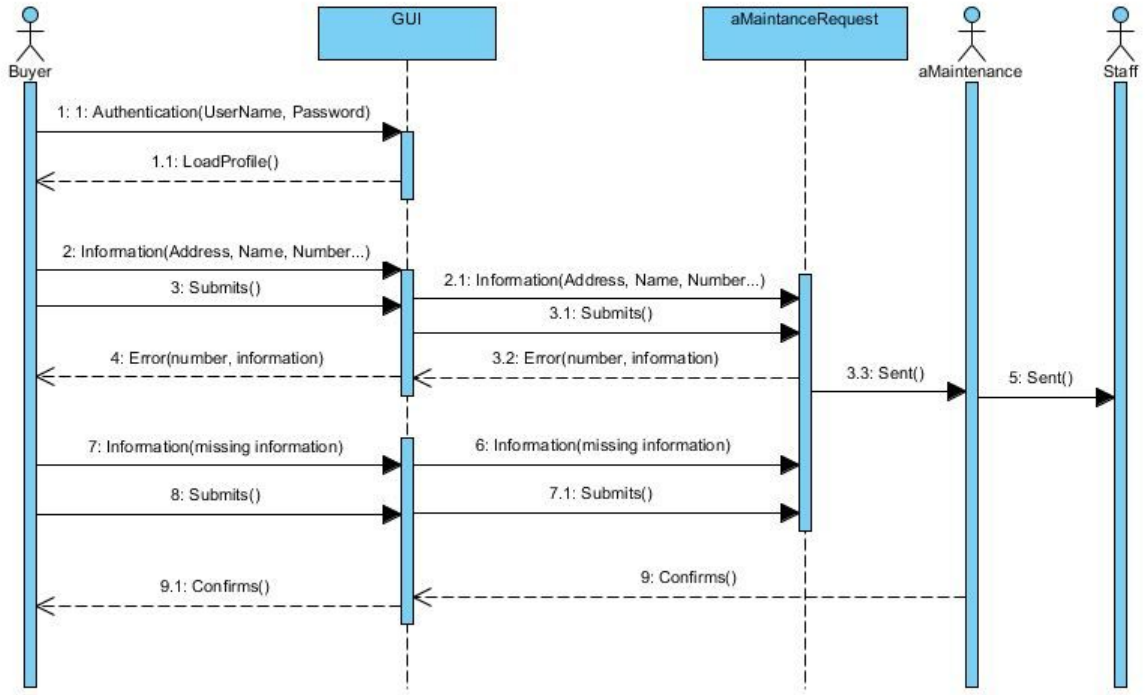
Author: Benya Chongolnee Date: 10/12/16

| | | |
|---|---|---|
| Use-Case Name: | Submit maintenance request | Use Case Type<br>Business Requirements: X<br>System Analysis:<br>o<br>System Design:<br>o |
| Use-Case ID: | 007 | |
| Priority: | Med | |
| Source: | Backlog | |
| primary business actor: | Buyer and maintenance | |
| Primary System Actor: | Buyer and maintenance | |
| Other Participating Actors: | Landlord and staff | |
| Other Interested Stockholders: | Landlord | |
| Description: | Buyer are able to submit maintenance request which then can be confirm/deny by the maintenance | |
| Precondition: | Buyer already bought/rent a house | |
| Trigger: | Buyer clicks on “maintenance request” on their homepage. | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1) Buyer clicks on “maintenance | 2) The page loads with |

| | | |
|---|---|---|
| | request" on their homepage<br><br><br><br>3) User fills in all of the blanks<br>4) User clicks submit | information for uses to fill in such as home address, name, maintenance type, time availability, contact number, and notes section<br><br><br><br>5) The system saves the information<br>6) The system sends this information to the maintenance record and the landlord<br>7) The system send a confirmation to the user |
| Alternate Courses: | 3a) The user does not fill on all of the blanks, so there will be an error message that will pops up.<br>4a) The user no longer wants to submit request so they clicks cancel | |
| Conclusion: | The buyer is able to submit maintenance request that will be able to be accessed by the maintenance | |
| Postcondition: | Return user to home page | |
| Business Rules: | none | |
| Impl. Constraints and Specifications: | none | |
| Assumptions: | The user puts in an existing house that is in the database | |
| Open Issues: | none | |


Submit Maintenance Request
Enter housing information
Validate all information is filled out
All info filled out?
Yes
No
Was cancel clicked?
No
Yes
Tell user to fill out all required data
Save maintenance request
Send maintenance request to maintenance record and landlord
Send confirmation to user

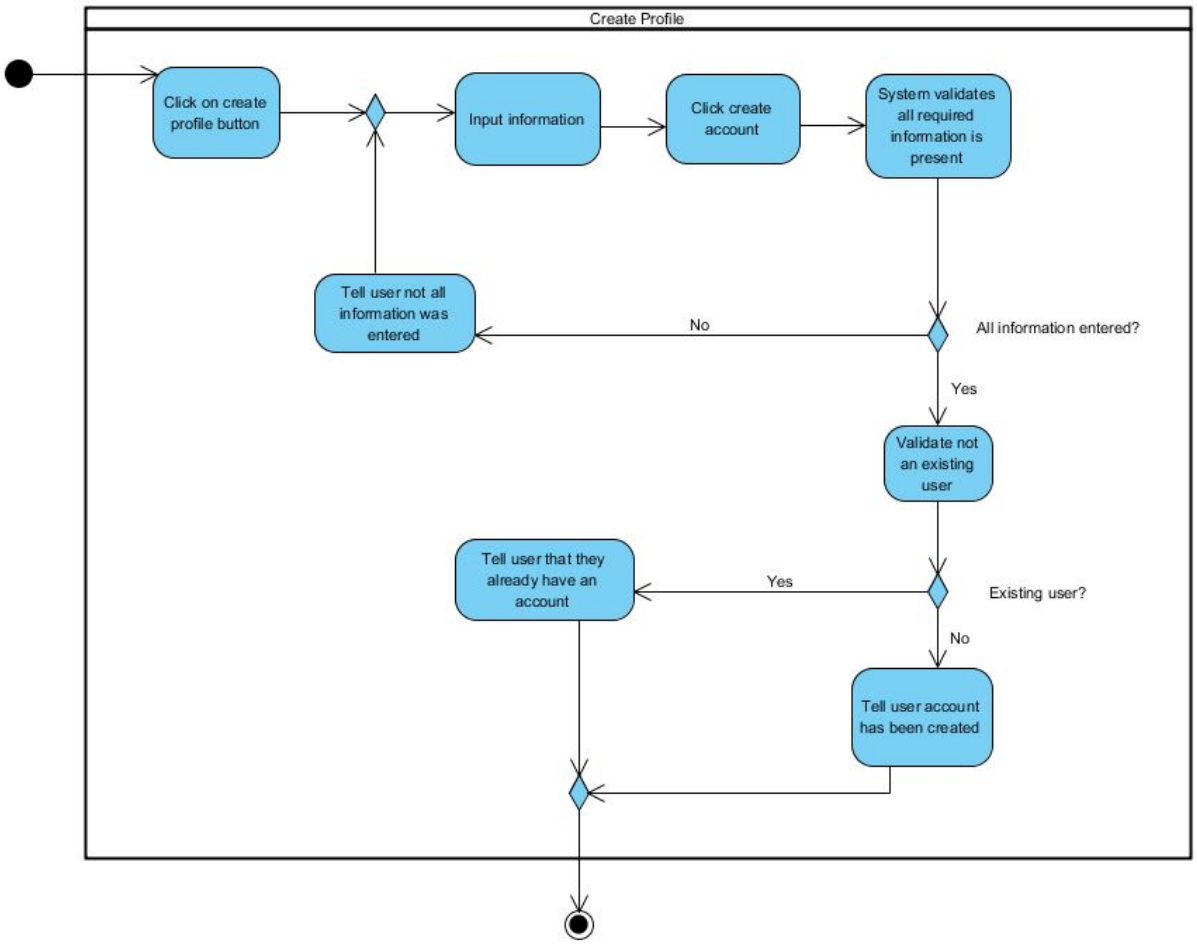

sd Submit Maintenance Request
Buyer
GUI
aProfile
aMaintanceRequest
aMaintenance
aStaff
1: Authenticate(Username, Password)
1.1: Confirm()
2: Sent()
3: Information(Address, Name, Number...)
4: Submit
4.1: Confirm()
5: Sent()

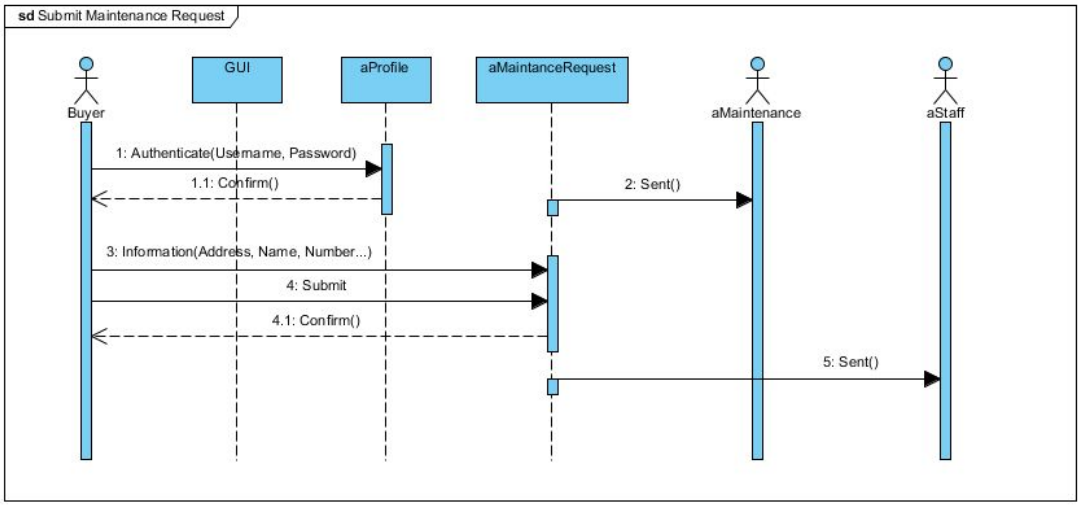

Buyer
GUI
aMaintanceRequest
aMaintenance
Staff
1: 1: Authentication(UserName, Password)
1.1: LoadProfile()
2: Information(Address, Name, Number...)
3: Submits()
2.1: Information(Address, Name, Number...)
3.1: Submits()
4: Error(number, information)
3.2: Error(number, information)
3.3: Sent()
5: Sent()
7: Information(missing information)
6: Information(missing information)
8: Submits()
7.1: Submits()
9.1: Confirms()
9: Confirms()

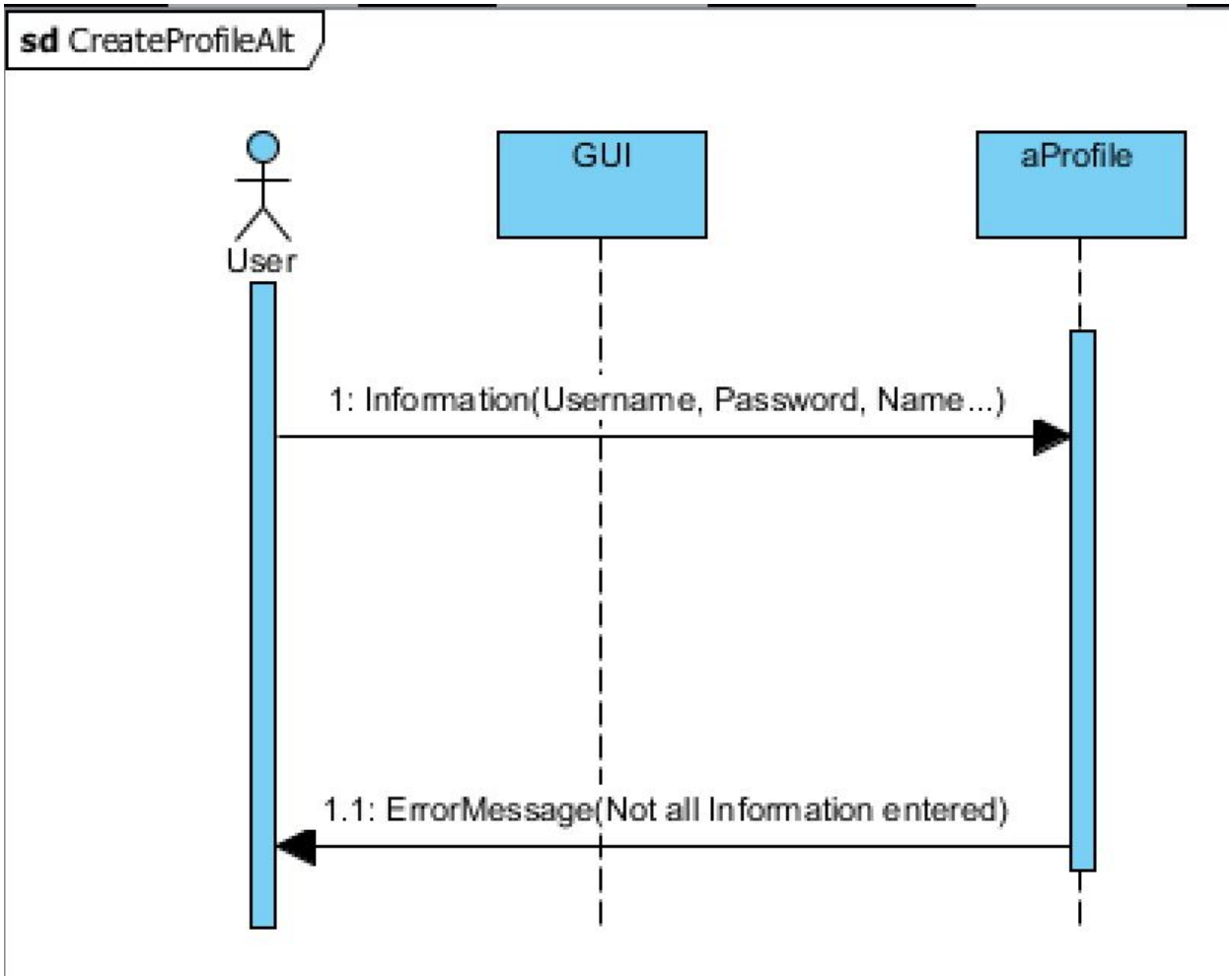
Author: Joshua Armstrong Date: 09/20/2016

| | | |
|---|---|---|
| Use-Case Name: | Create Profile | Use Case Type<br>Business Requirements: [x]<br>System Analysis: o<br>System Design: o |
| Use-Case ID: | 010 | |
| Priority: | High | |
| Source: | Backlog | |
| primary business actor: | Buyer | |
| Primary System Actor: | Buyer | |
| Other Participating Actors: | n/a | |
| Other Interested Stockholders: | Staff, Sellers | |
| Description: | Buyer creates a profile to see what houses match their interests | |
| Precondition: | Access Website | |
| Trigger: | User click create profile button | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1. User clicks the “create account” button in the login page | 2. “Create account” page loads |

| | | |
|---|---|---|
| | 3. User inputs Information<br>4. User clicks creat<br><br>6. User logs in with new username and password | 5. System Validates no existing user, and information |
| Alternate Courses: | 5a) Error message informing user that not all required information has not been entered<br>1 | |
| Conclusion: | System creates new user account | |
| Postcondition: | Prompt user to sign in | |
| Business Rules: | Only one account per unique Email | |
| Impl. Constraints and Specifications: | Email address must not already have an existing account | |
| Assumptions: | User is a new Customer, and has an email address | |
| Open Issues: | None | |


Create Profile
Click on create profile button
Input information
Click create account
System validates all required information is present
Tell user not all information was entered
No
All information entered?
Yes
Validate not an existing user
Tell user that they already have an account
Yes
Existing user?
No
Tell user account has been created

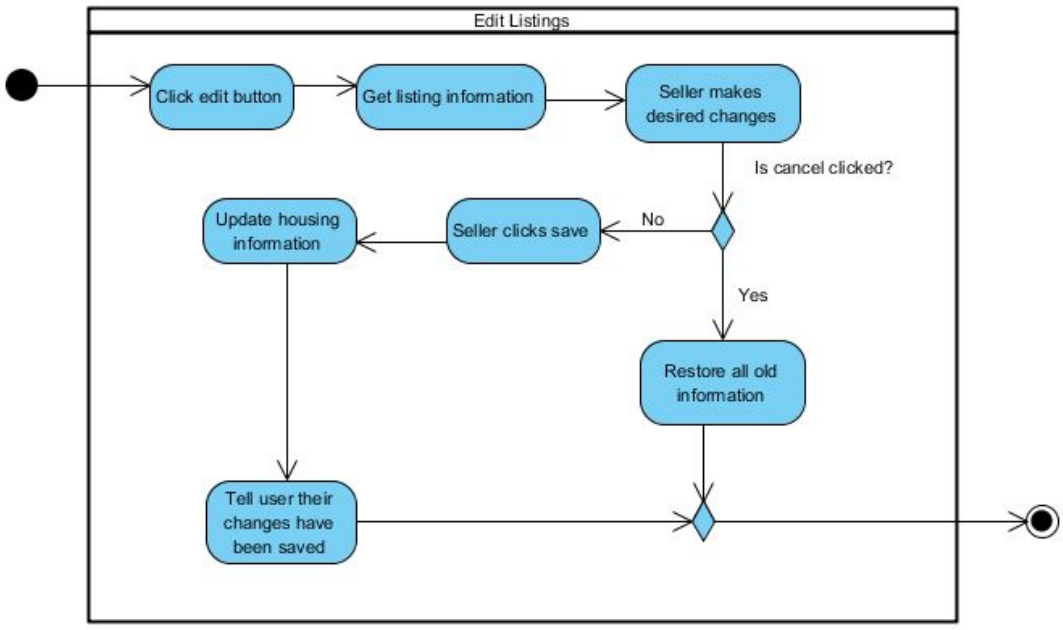

sd Submit Maintenance Request
Buyer
GUI
aProfile
aMaintanceRequest
aMaintenance
aStaff
1: Authenticate(Username, Password)
1.1: Confirm()
2: Sent()
3: Information(Address, Name, Number...)
4: Submit
4.1: Confirm()
5: Sent()

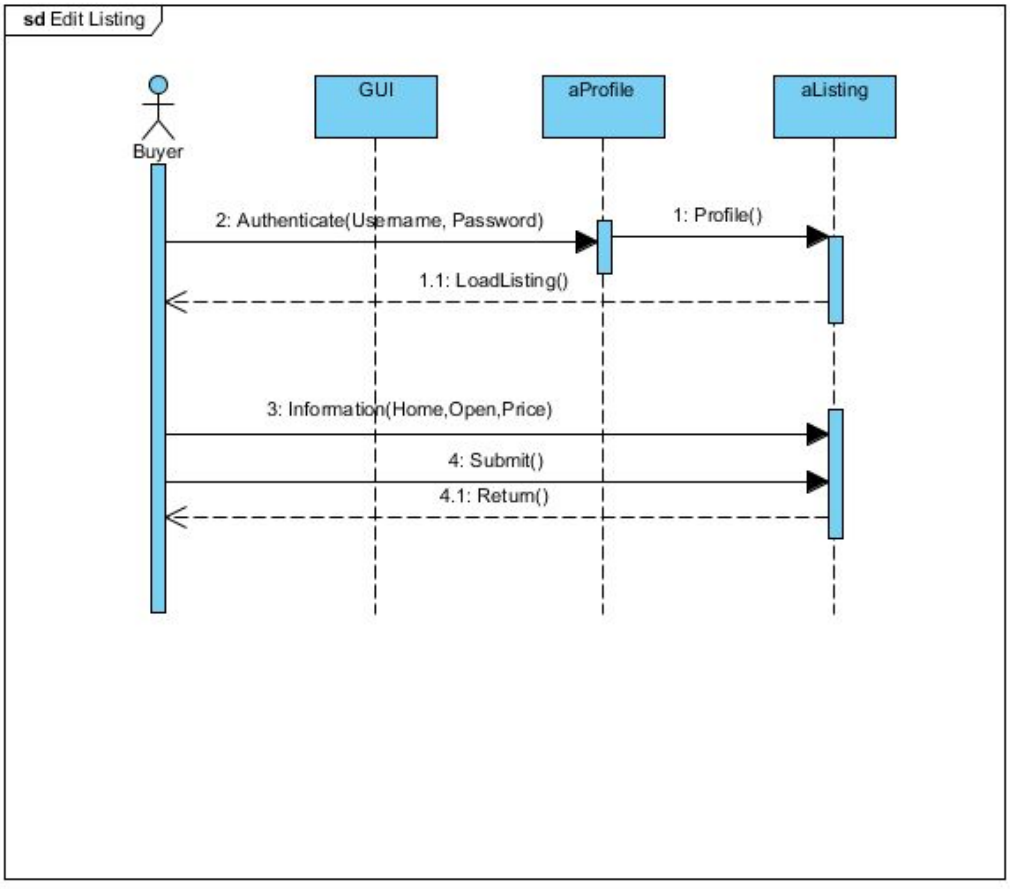

sd CreateProfileAlt
User
GUI
aProfile
1: Information(Username, Password, Name...)
1.1: ErrorMessage(Not all Information entered)

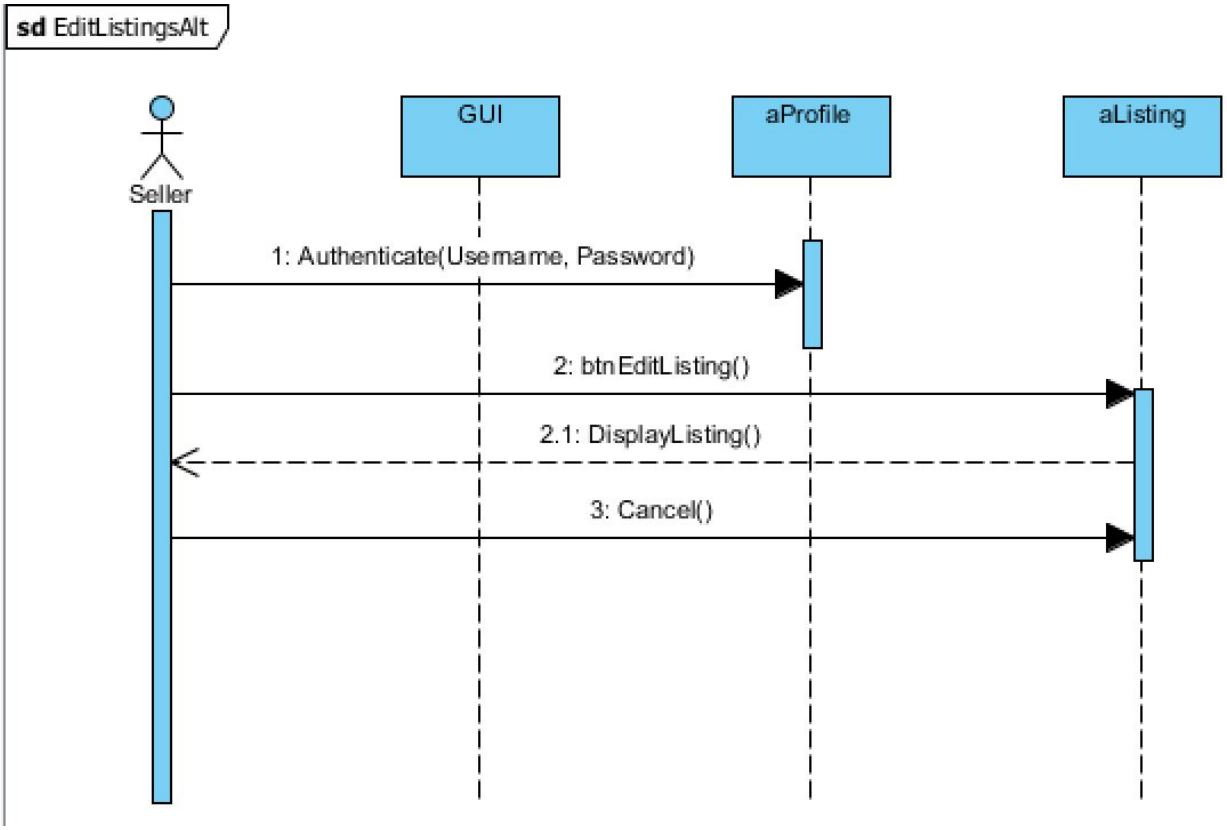
Author: Joshua Armstrong Date: 09/20/2016

| | | |
|---|---|---|
| Use-Case Name: | Edit Listings | Use Case Type<br>Business Requirements: [x]<br>System Analysis:<br>o<br>System Design:<br>o |
| Use-Case ID: | 014 | |
| Priority: | High | |
| Source: | Backlog | |
| primary business actor: | Buyer | |
| Primary System Actor: | Seller | |
| Other Participating Actors: | None | |
| Other Interested Stockholders: | Staff | |
| Description: | Seller can edit the information on their listings | |
| Precondition: | The seller already has a listing in the Home Sweet Home Business System | |
| Trigger: | The seller decides what listing to edit, then clicks the “Edit” button | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1. Seller clicks “edit listing” | 2. System pulls up the listing |

| | | |
|---|---|---|
| | | info |
| | 3. Seller edits/changes desired information<br>4. Seller clicks the save button | |
| | | 5. System saves the new information for the listing.<br>6. Prompt seller informing them that the listing has been updated |
| Alternate Courses: | 4a) Seller changes mind, and clicks cancel, recycling all new information | |
| Conclusion: | New information is updated for a listing | |
| Postcondition: | Return user to the User Homepage | |
| Business Rules: | | |
| Impl. Constraints and Specifications: | none | |
| Assumptions: | Seller already has a home listed | |
| Open Issues: | none | |


Edit Listings
Click edit button
Get listing information
Seller makes desired changes
Is cancel clicked?
No
Seller clicks save
Update housing information
Yes
Restore all old information
Tell user their changes have been saved

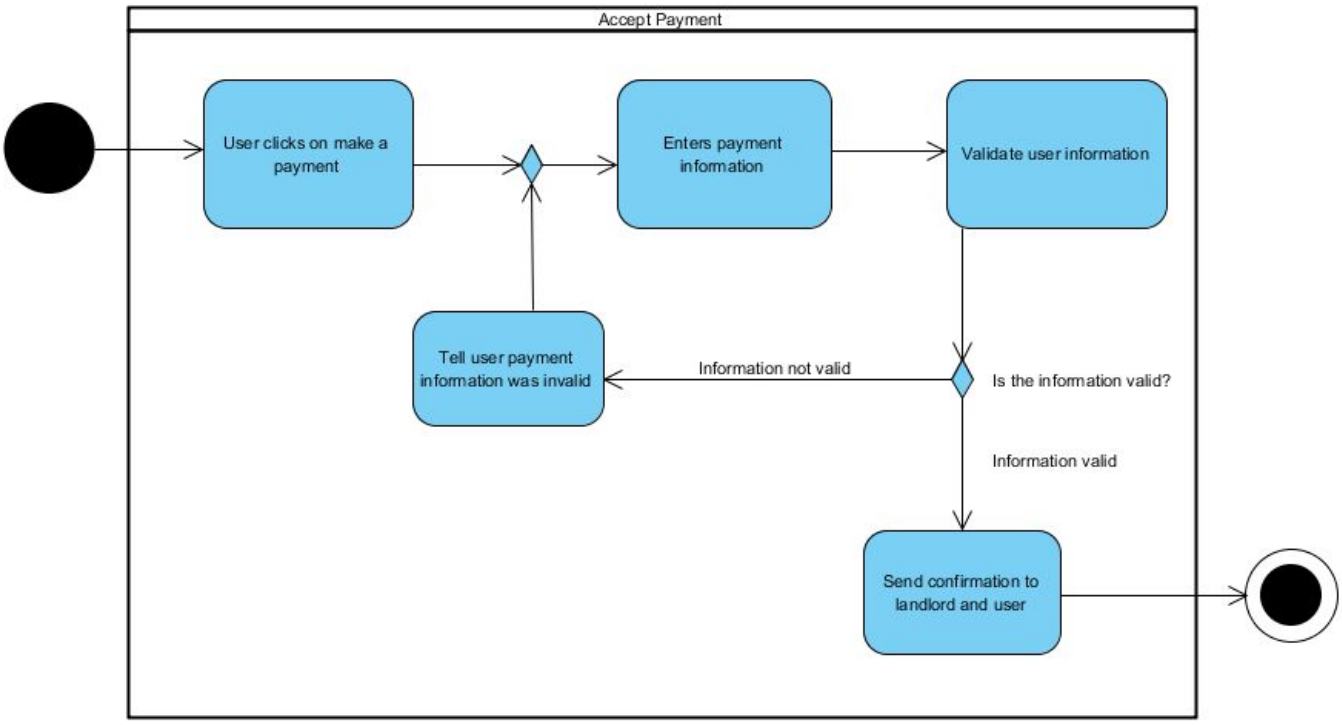

sd Edit Listing
Buyer
GUI
aProfile
aListing
2: Authenticate(Username, Password)
1: Profile()
1.1: LoadListing()
3: Information(Home,Open,Price)
4: Submit()
4.1: Return()

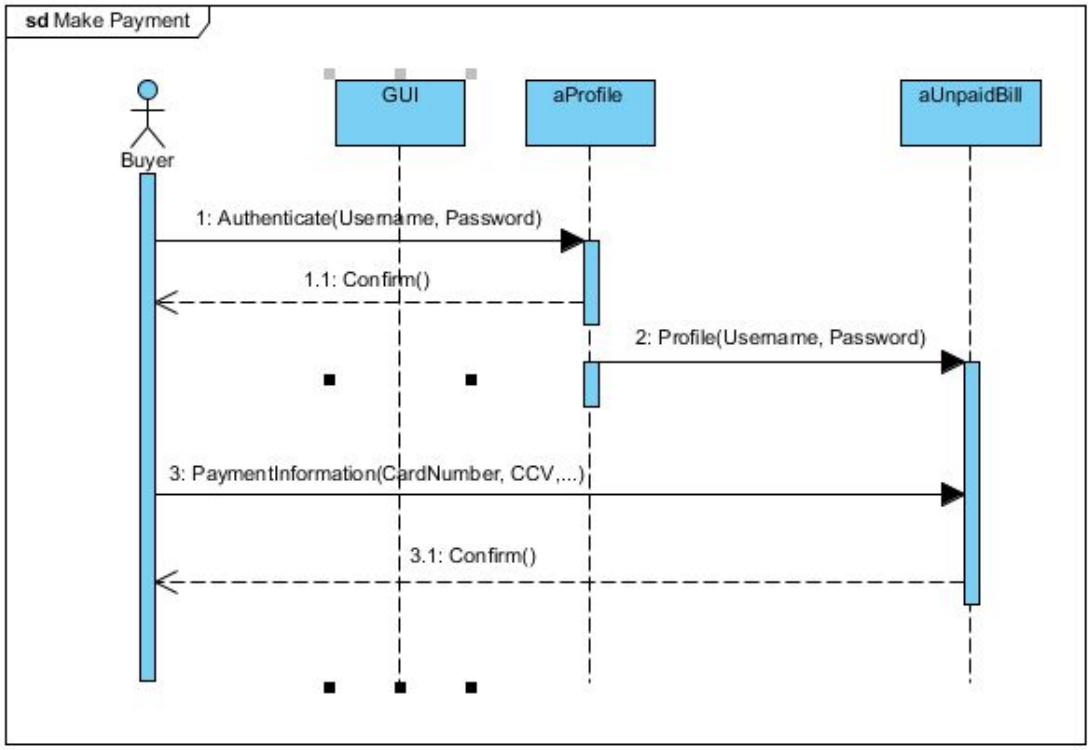

sd EditListingsAlt
Seller
GUI
aProfile
aListing
1: Authenticate(Username, Password)
2: btnEditListing()
2.1: DisplayListing()
3: Cancel()

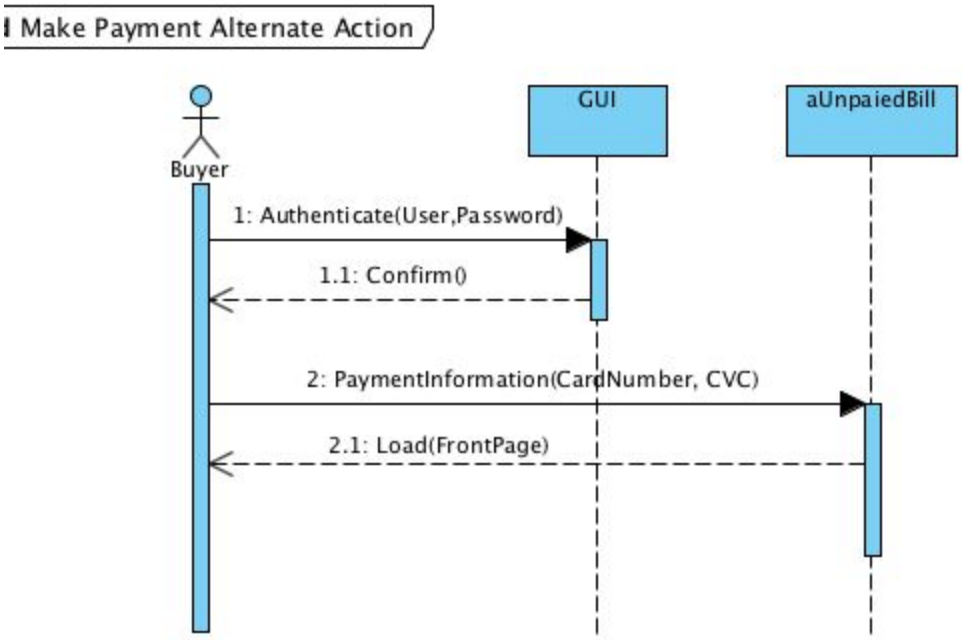
Author: Mathew Berry Date:9/8/16

| | | |
|---|---|---|
| Use-Case Name: | Accept Payment | Use Case Type<br>Business Requirements: X<br>System Analysis: o<br>System Design: o |
| Use-Case ID: | 002 | |
| Priority: | Mid | |
| Source: | User requirements | |
| primary business actor: | Seller | |
| Primary System Actor: | Buyer | |
| Other Participating Actors: | LLMS Staff | |
| Other Interested Stockholders: | None | |
| Description: | User paying the landlord rent money | |
| Precondition: | User has to be a tenant | |
| Trigger: | User clicks on make payment | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1) Clicks on make payment<br>3) User enters payment information | 2) Front page loaded<br>4) System validates user information<br>5) Send confirmation to landlord and user |
| Alternate Courses: | 3a)User enters invalid information<br>3b) System returns to step 2 | |
| Conclusion: | Landlord receives rent money | |

| | |
|---|---|
| Postcondition: | Payment approved |
| Business Rules: | Does not accept cash |
| Impl. Constraints and Specifications: | If late don't accept payment |
| Assumptions: | Landlord doesn't take cash |
| Open Issues: | None |



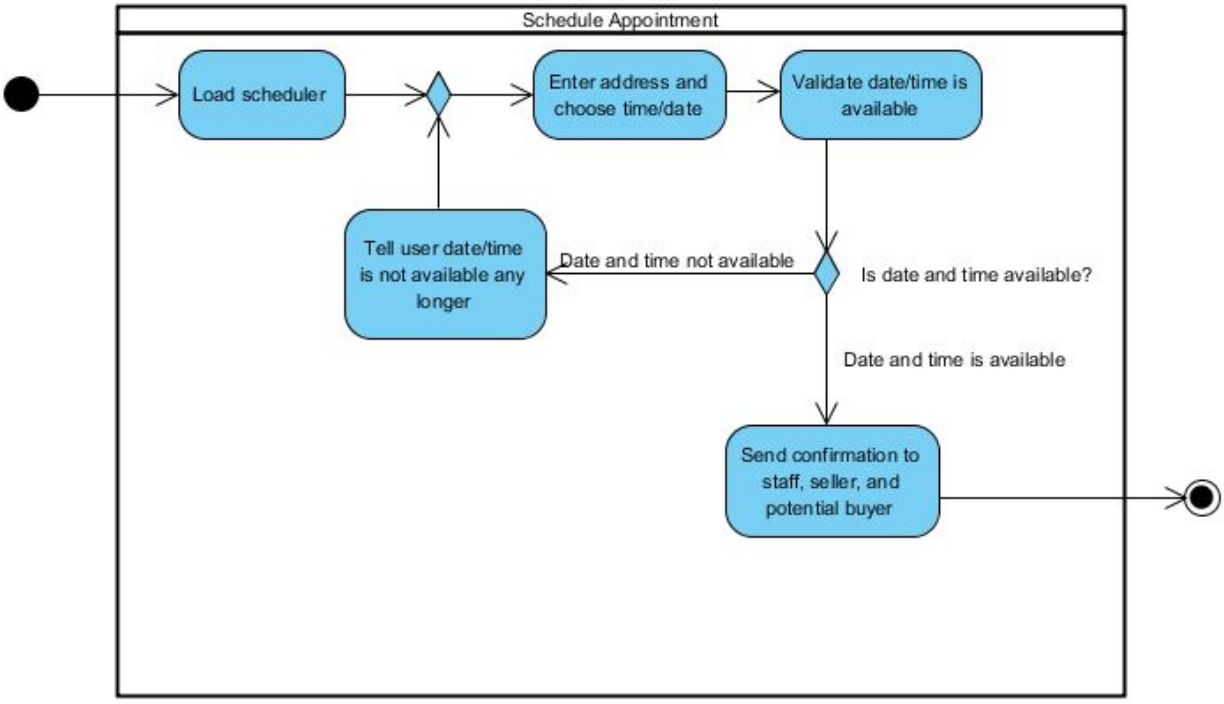

sd Make Payment
Buyer
GUI
aProfile
aUnpaidBill
1: Authenticate(Username, Password)
1.1: Confirm()
2: Profile(Username, Password)
3: PaymentInformation(CardNumber, CCV,...)
3.1: Confirm()

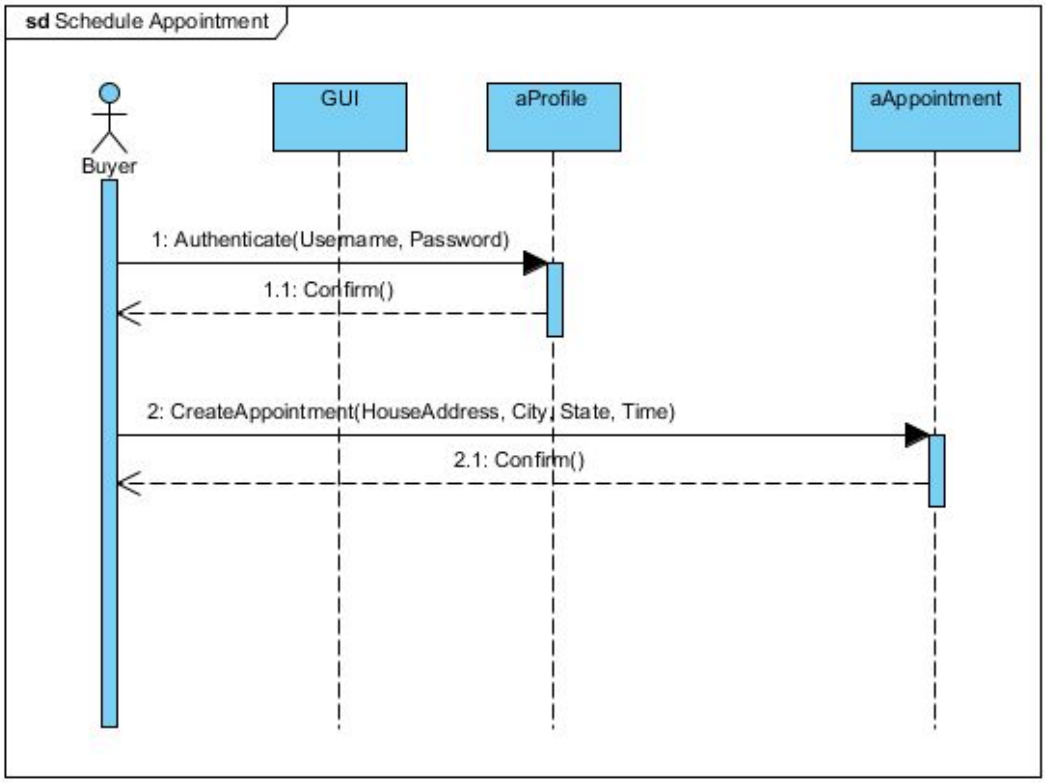

Make Payment Alternate Action
Buyer
GUI
aUnpaiedBill
1: Authenticate(User,Password)
1.1: Confirm()
2: PaymentInformation(CardNumber, CVC)
2.1: Load(FrontPage)

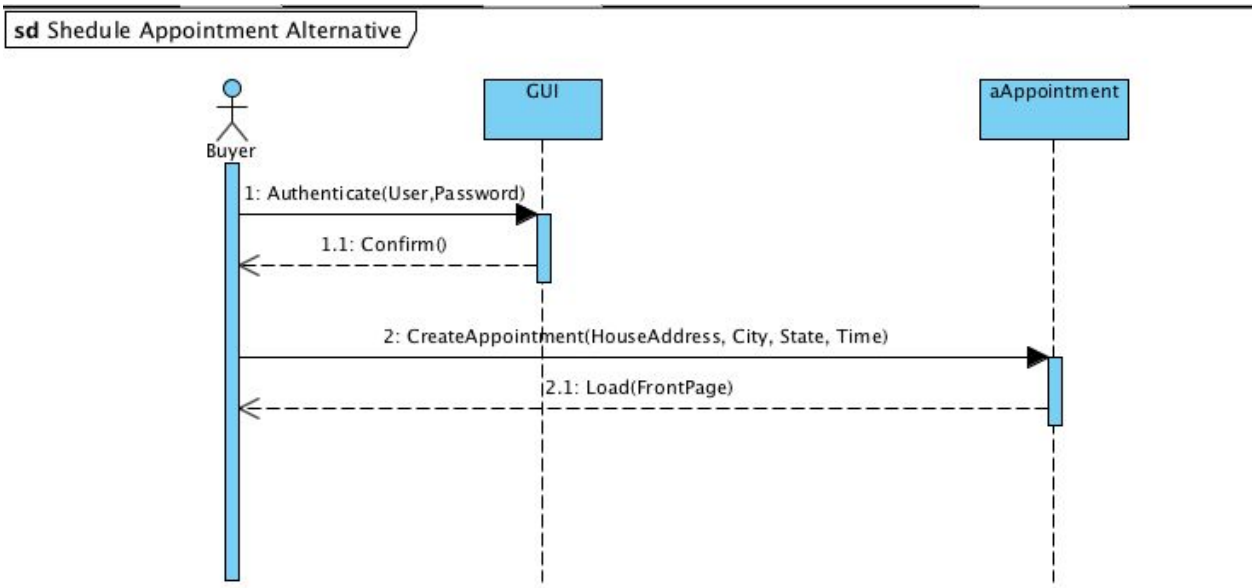
Author: Mathew Berry Date: 9/8/16

| | | |
|---|---|---|
| Use-Case Name: | Schedule Appointment | Use Case Type<br>Business Requirements: X<br>System Analysis:<br>o<br>System Design:<br>o |
| Use-Case ID: | 016 | |
| Priority: | Med | |
| Source: | User Requirements | |
| primary business actor: | Seller | |
| Primary System Actor: | Buyer | |
| Other Participating Actors: | House owner, Seller's realtor | |
| Other Interested Stockholders: | None | |
| Description: | LLMS staff schedules an appointment for the client | |
| Precondition: | Client must be a potential buyer | |
| Trigger: | Clicks on "make appointment" | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1) Buyer clicks on "make appointment" | 2) Front page loaded |

| | | |
|---|---|---|
| | 3) Buyer enters address and chooses time/date<br><br>4) Buy clicks submit | <br><br><br><br>5) Validate date/time is available<br><br>6) Sends confirmation to staff member, seller and potential buyer |
| Alternate Courses: | 3a) date/time is not available<br>3b) System returns to step 1 | |
| Conclusion: | Appointment is scheduled | |
| Postcondition: | None | |
| Business Rules: | A house can have multiple appointments in one day | |
| Impl. Constraints and Specifications: | None | |
| Assumptions: | None | |
| Open Issues: | None | |


Schedule Appointment
Load scheduler
Enter address and choose time/date
Validate date/time is available
Tell user date/time is not available any longer
Date and time not available
Is date and time available?
Date and time is available
Send confirmation to staff, seller, and potential buyer

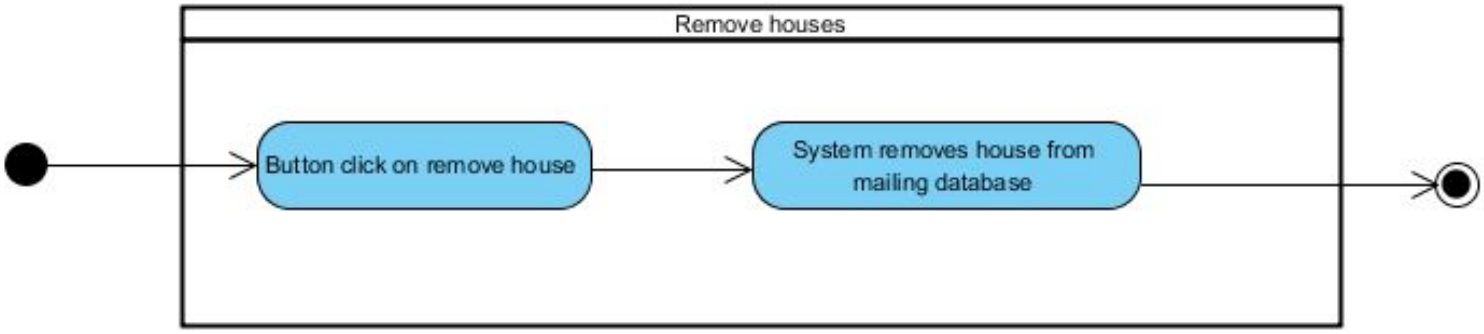

sd Schedule Appointment
Buyer
GUI
aProfile
aAppointment
1: Authenticate(Username, Password)
1.1: Confirm()
2: CreateAppointment(HouseAddress, City, State, Time)
2.1: Confirm()

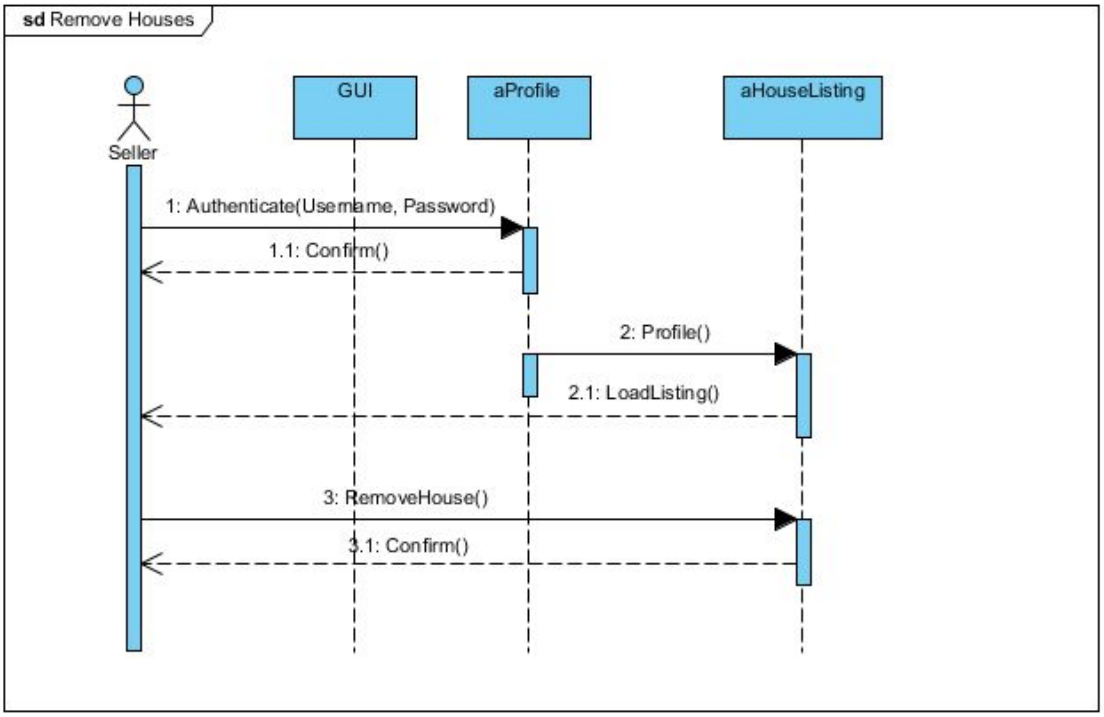

sd Shedule Appointment Alternative
Buyer
GUI
aAppointment
1: Authenticate(User,Password)
1.1: Confirm()
2: CreateAppointment(HouseAddress, City, State, Time)
2.1: Load(FrontPage)

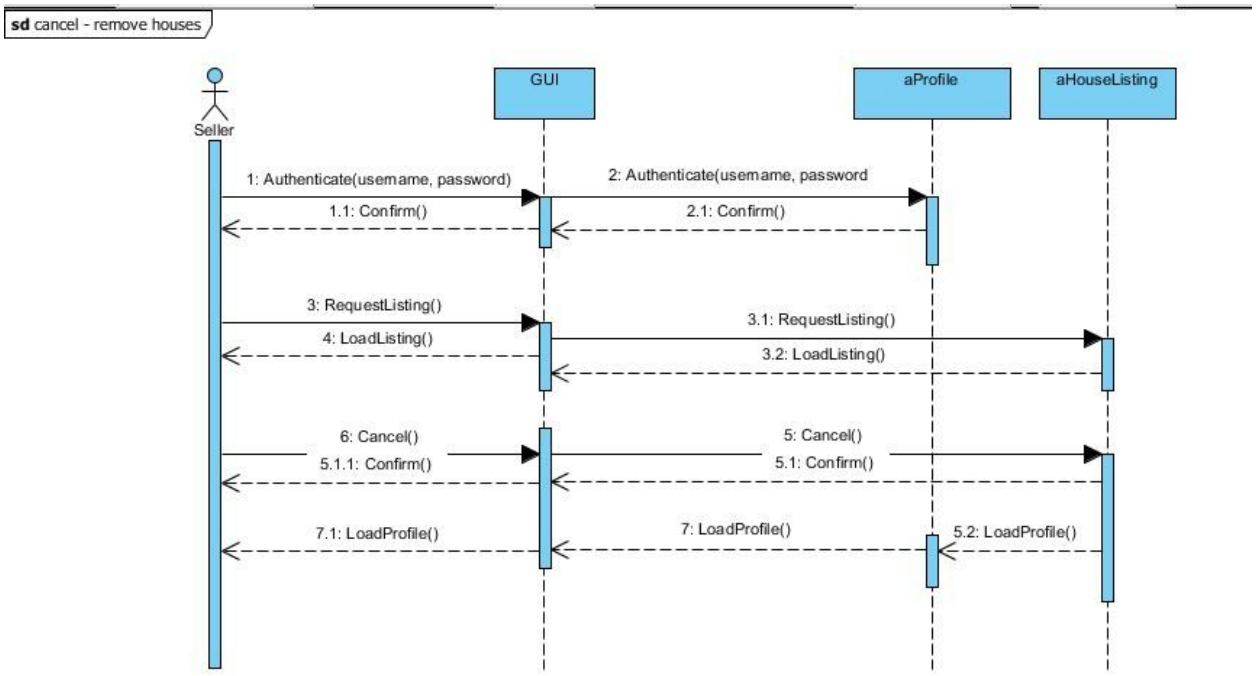
Author: Kyle Kmetz Date: 10/15/16

| | | |
|---|---|---|
| Use-Case Name: | Remove houses | Use Case Type<br>Business Requirements: □X |
| Use-Case ID: | 013 | |
| Priority: | Mid | System Analysis: □ |
| Source: | User requirements | System Design: □ |
| primary business actor: | Seller | |
| Primary System Actor: | Staff | |
| Other Participating Actors: | Buyer | |
| Other Interested Stockholders: | None | |
| Description: | Remove houses off the informational mailing list when they are no longer on the market | |
| Precondition: | House was in the mailing list | |
| Trigger: | Staff clicks on delete listing on staff page | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1) Staff clicks on delete listing on staff page | |
| | | 2) System loads available houses in the database |
| | 3) Staff click on the listing that they want it to be removed. | |
| | 4) Staff clicks delete button | |

| | | |
|---|---|---|
| | | 5) System deletes the house listing and notify the seller and the staff |
| Alternate Courses: | 3a) The user no longer wants to remove the house. Therefore the user clicks “Home”<br><br>3b) Go back to home page | |
| Conclusion: | House has been taken off the mailing list | |
| Postcondition: | Buyers will not get information on this specific house in the mailing list | |
| Business Rules: | Only staff and landlord can remove off the mailing list | |
| Impl. Constraints and Specifications: | None | |
| Assumptions: | User is allowed to communicate to the database. | |
| Open Issues: | None | |


Remove houses
Button click on remove house
System removes house from mailing database

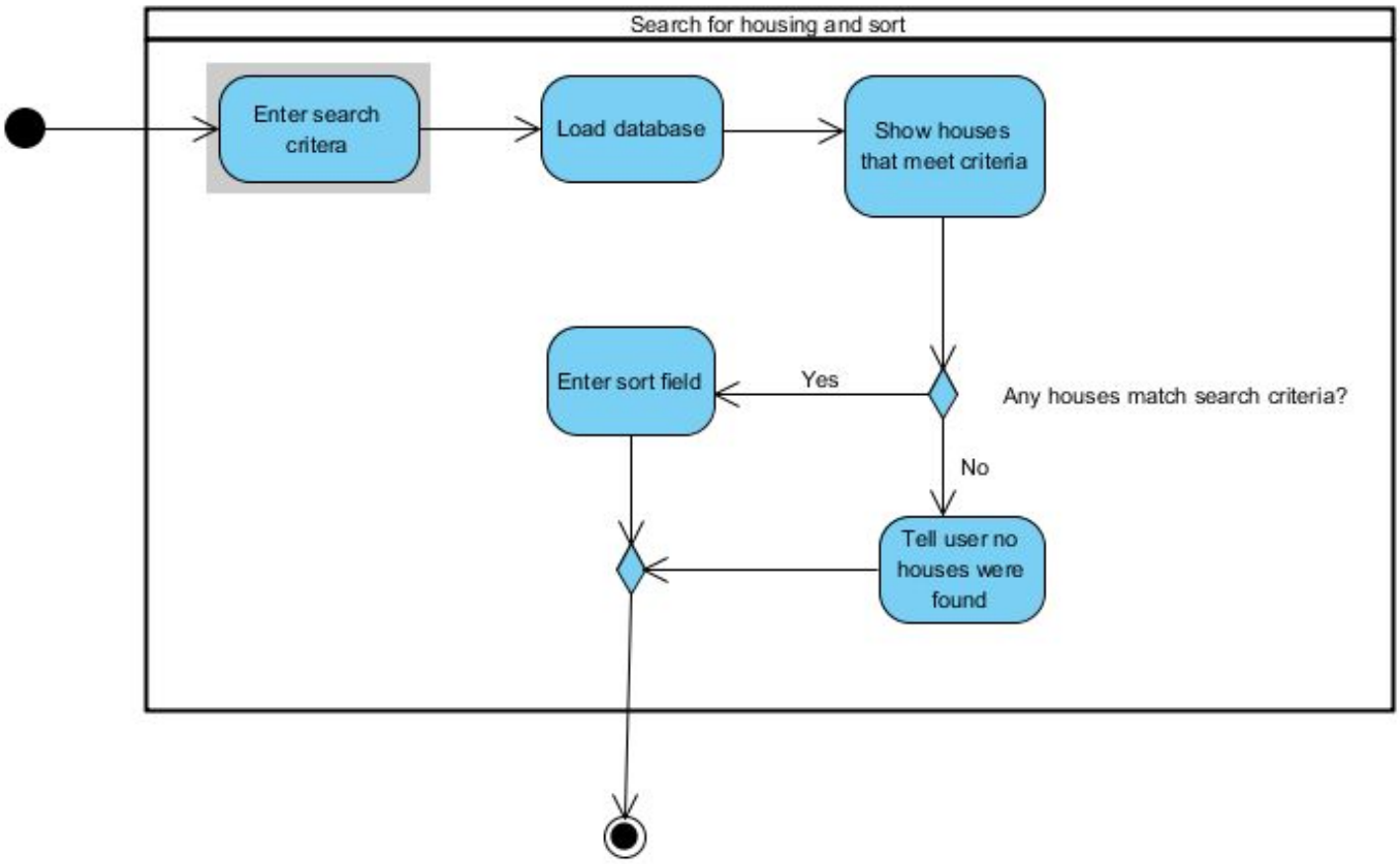

sd Remove Houses
Seller
GUI
aProfile
aHouseListing
1: Authenticate(Username, Password)
1.1: Confirm()
2: Profile()
2.1: LoadListing()
3: RemoveHouse()
3.1: Confirm()

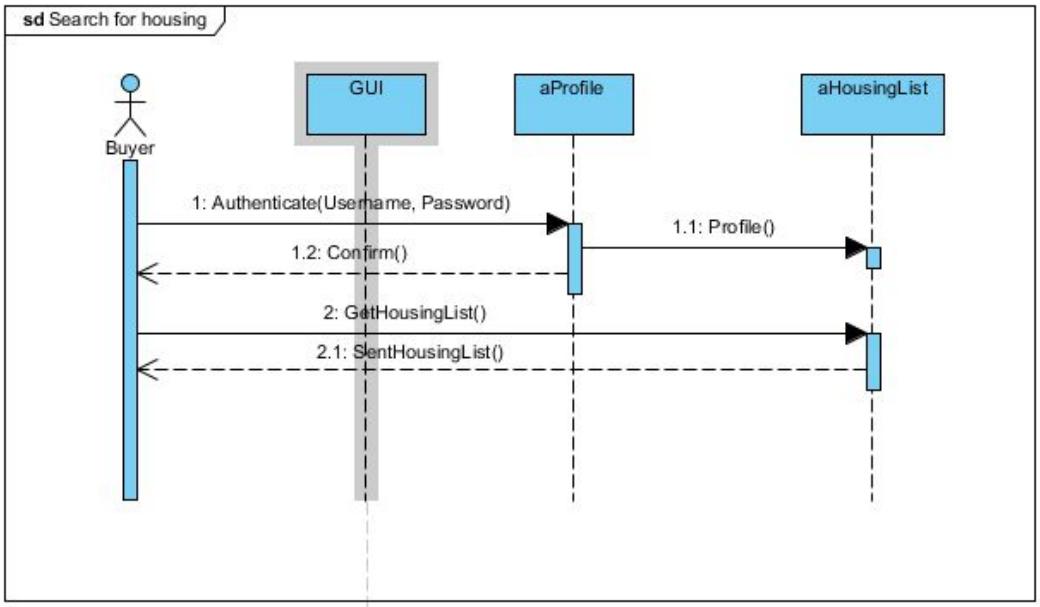

sd cancel - remove houses
Seller
GUI
aProfile
aHouseListing
1: Authenticate(username, password)
1.1: Confirm()
2: Authenticate(username, password
2.1: Confirm()
3: RequestListing()
4: LoadListing()
3.1: RequestListing()
3.2: LoadListing()
6: Cancel()
5.1.1: Confirm()
5: Cancel()
5.1: Confirm()
7.1: LoadProfile()
7: LoadProfile()
5.2: LoadProfile()

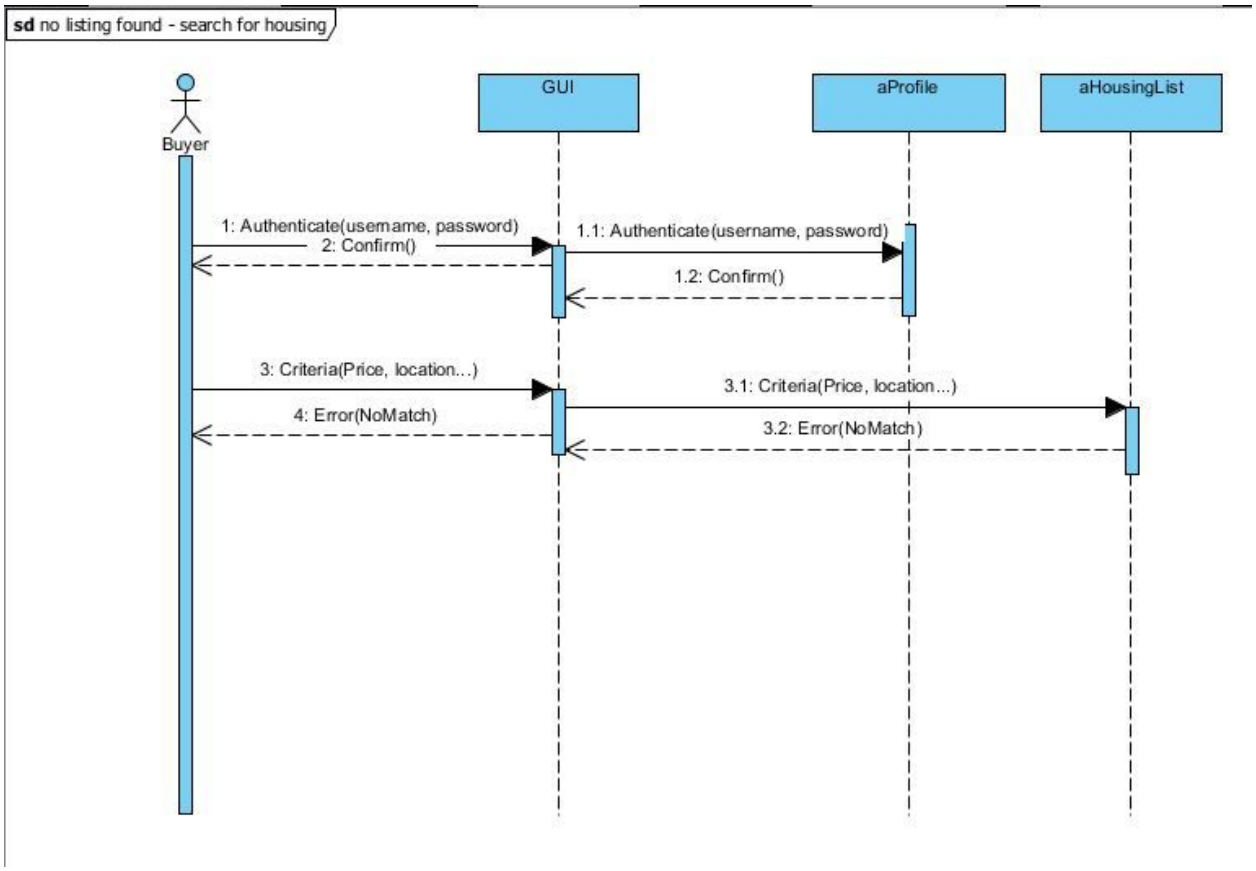
Author: Kyle Kmetz Date: 10/15/16

| | | |
|---|---|---|
| Use-Case Name: | Search for housing and sort | Use Case Type<br>Business Requirements: □X<br>System Analysis: □<br>System Design: □ |
| Use-Case ID: | 015 | |
| Priority: | Mid | |
| Source: | User requirements | |
| primary business actor: | Buyer | |
| Primary System Actor: | Buyer | |
| Other Participating Actors: | LLMS Staff | |
| Other Interested Stockholders: | Seller | |
| Description: | User searches for housing criteria and sorts by certain specification | |
| Precondition: | There are housing option to search for | |
| Trigger: | User clicks on search listings | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1) User clicks on search listings<br><br>3)Enters search criteria and clicks search | 2) Search page loads<br><br>4) System finds housing data that fits criteria<br>5) System shows the houses that fits the criteria |

| | | |
|---|---|---|
| | 6) User clicks view to tour the house in the person | 7) System loads “make appointment” form |
| Alternate Courses: | 3a) System finds no housing with criteria<br>3b) system tells the user that no results were found | |
| Conclusion: | Buyer can view specific data in a customizable order | |
| Postcondition: | N/A | |
| Business Rules: | If no sort order is chosen, data output is sorted by name | |
| Impl. Constraints and Specifications: | If house has been taken off the market, dont show it | |
| Assumptions: | All houses have all data filled in | |
| Open Issues: | None | |


Search for housing and sort
Enter search critera
Load database
Show houses that meet criteria
Any houses match search criteria?
Yes
Enter sort field
No
Tell user no houses were found

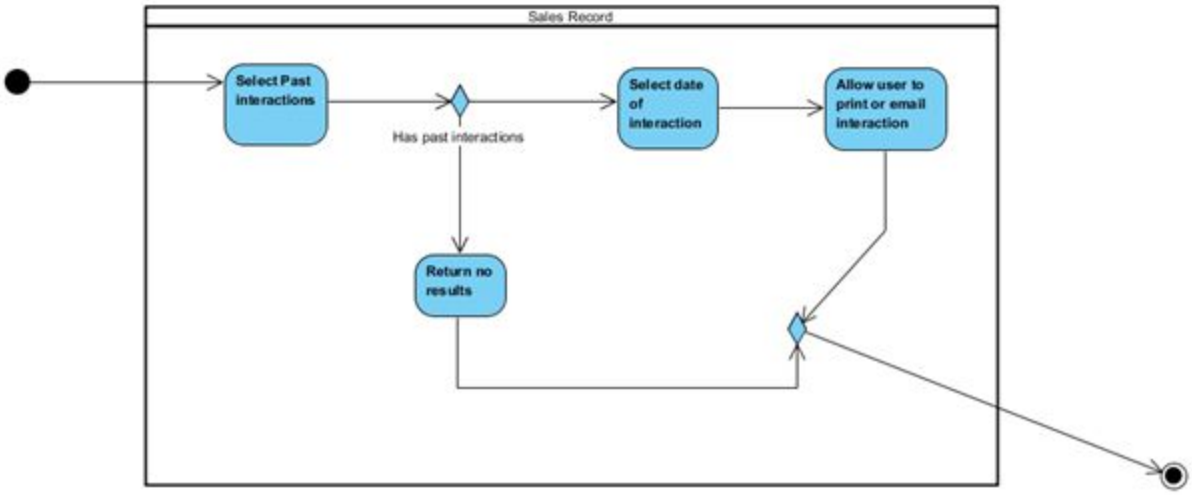

sd Search for housing
Buyer
GUI
aProfile
aHousingList
1: Authenticate(Username, Password)
1.1: Profile()
1.2: Confirm()
2: GetHousingList()
2.1: SentHousingList()

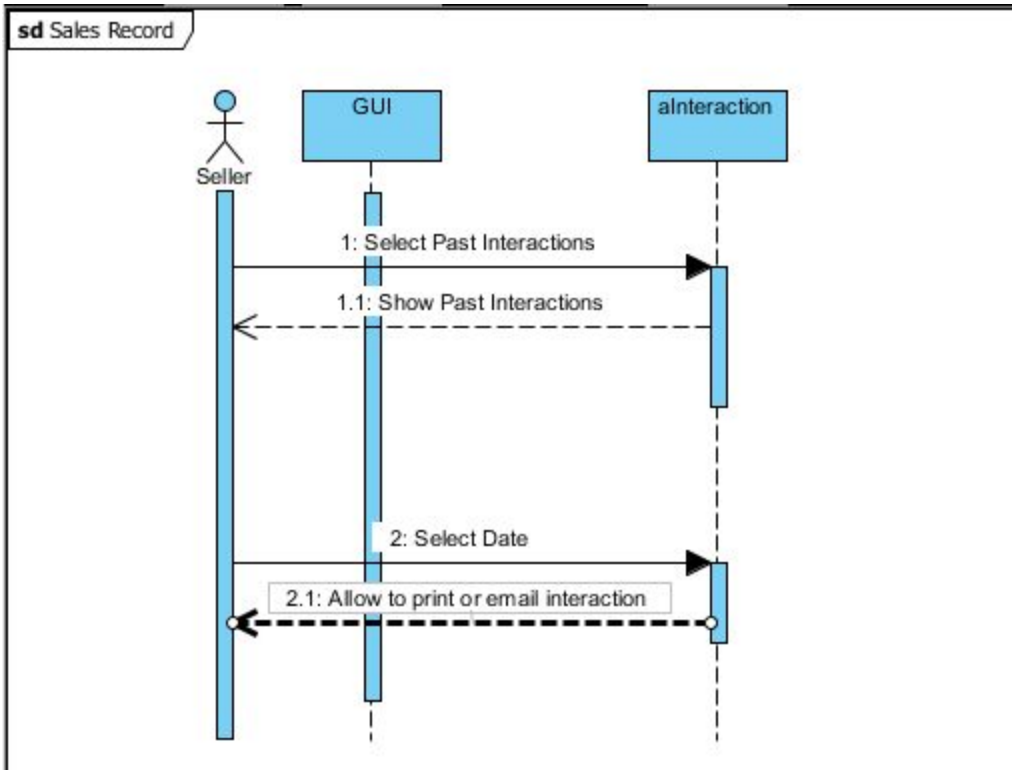

sd no listing found - search for housing
Buyer
GUI
aProfile
aHousingList
1: Authenticate(username, password)
2: Confirm()
1.1: Authenticate(username, password)
1.2: Confirm()
3: Criteria(Price, location...)
4: Error(NoMatch)
3.1: Criteria(Price, location...)
3.2: Error(NoMatch)

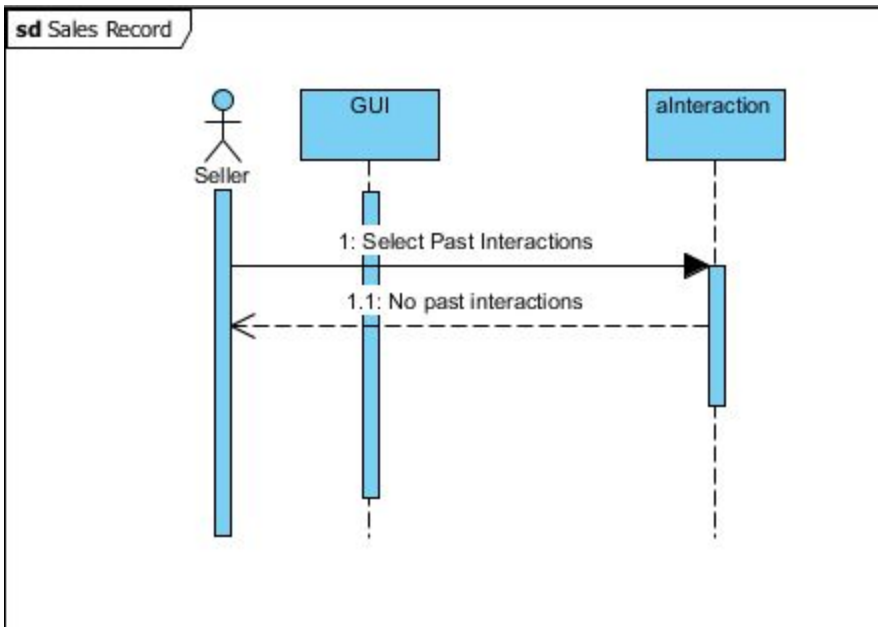
Author: Joey Mucha Date: 10/19/16

| | | |
|---|---|---|
| Use-Case Name: | Sales Record | Use Case Type<br>Business Requirements: □x |
| Use-Case ID: | 017 | |
| Priority: | Low | System Analysis: □ |
| Source: | Seller Requirements | System Design: □ |
| primary business actor: | Seller | |
| Primary System Actor: | Staff | |
| Other Participating Actors: | None | |
| Other Interested Stockholders: | None | |
| Description: | Sales are recorded so the seller can keep track in case of legal issues. | |
| Precondition: | House must be sold | |
| Trigger: | Seller selects past interactions | |
| Typical Course Of Events: | Actor Action | System Response |
| | 1) Seller selects past interactions<br>3) Seller selects a date the desired past interaction<br>4) User selects “load” | 2) System loads past interactions |

| | | |
|---|---|---|
| | 5) User clicks Email | 4) System loads the transaction during the specified date.<br><br>6) System sends the email to the staff with all of the information<br><br>7) Confirms with user |
| Alternate Courses: | 2a) Customer has no past interactions, system returns no results<br><br>1a) User no longer wants to find the history, user clicks “home” | |
| Conclusion: | House is added to sales record | |
| Postcondition: | Sale is recorded | |
| Business Rules: | Only seller can view the houses that they have sold | |
| Impl. Constraints and Specifications: | None | |
| Assumptions: | Seller has had past interaction with HSH | |
| Open Issues: | None | |


Sales Record
Select Past interactions
Has past interactions
Select date of interaction
Allow user to print or email interaction
Return no results

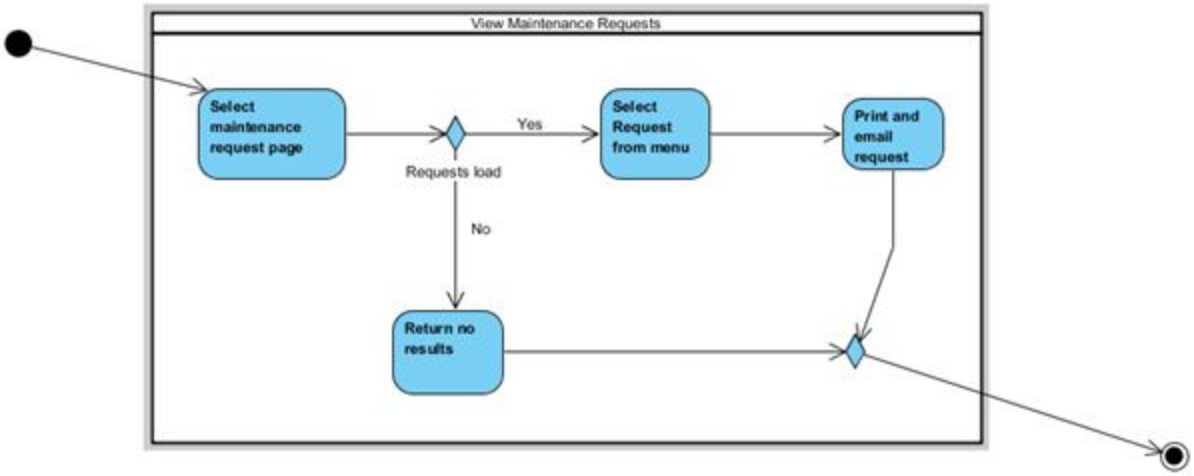

sd Sales Record
Seller
GUI
aInteraction
1: Select Past Interactions
1.1: Show Past Interactions
2: Select Date
2.1: Allow to print or email interaction

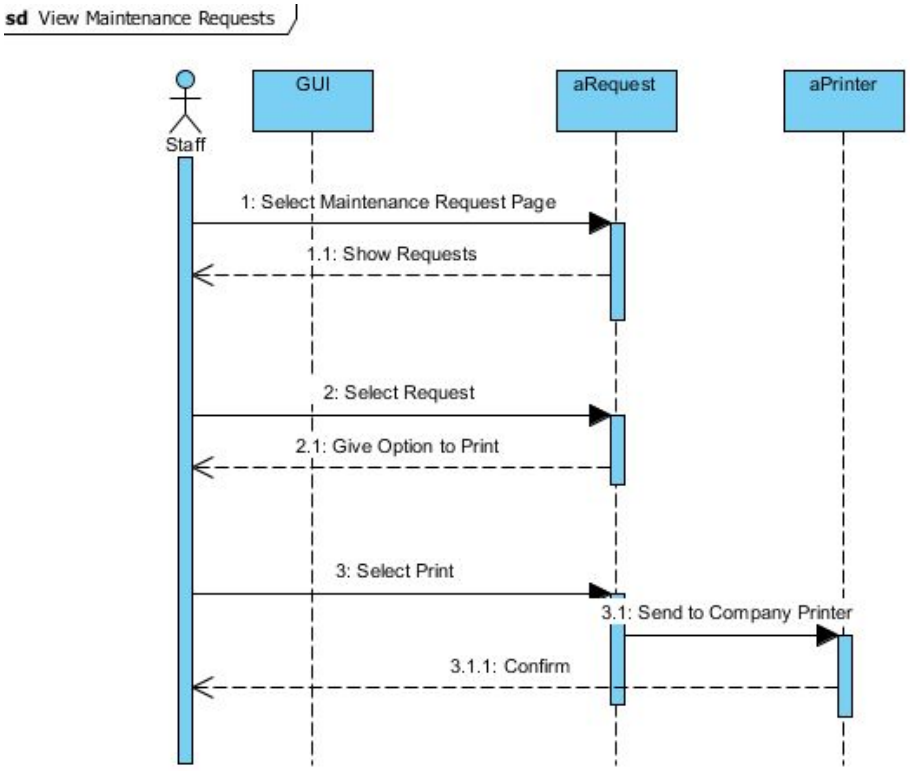

sd Sales Record
Seller
GUI
aInteraction
1: Select Past Interactions
1.1: No past interactions

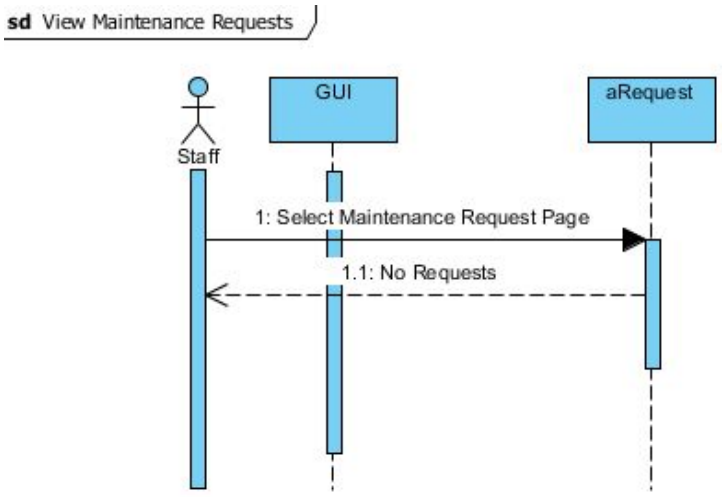
Author: Joey Mucha Date: 10/19/16

<table>
<tr><td>Use-Case Name:</td><td>View maintenance requests</td><td rowspan="4">Use Case Type<br><br>Business Requirements: □x<br><br>System Analysis: □<br><br>System Design: □</td></tr>
<tr><td>Use-Case ID:</td><td>018</td></tr>
<tr><td>Priority:</td><td>Mid</td></tr>
<tr><td>Source:</td><td>User Requirements</td></tr>
<tr><td>primary business actor:</td><td colspan="2">Maintenance Staff</td></tr>
<tr><td>Primary System Actor:</td><td colspan="2">User</td></tr>
<tr><td>Other Participating Actors:</td><td colspan="2">None</td></tr>
<tr><td>Other Interested Stockholders:</td><td colspan="2">None</td></tr>
<tr><td>Description:</td><td colspan="2">Maintenance staff must be able to view maintenance requests</td></tr>
<tr><td>Precondition:</td><td colspan="2">Request must be placed</td></tr>
<tr><td>Trigger:</td><td colspan="2">Maintenance Staff clicks “view requests”</td></tr>
<tr><td rowspan="2">Typical Course Of Events:</td><td>Actor Action</td><td>System Response</td></tr>
<tr><td>1) Maintenance staff selects “view requests”<br><br>3) Staff selects a request from a dropdown menu and selects load</td><td>2) Generates requests<br><br>4) System loads the request information</td></tr>
</table>

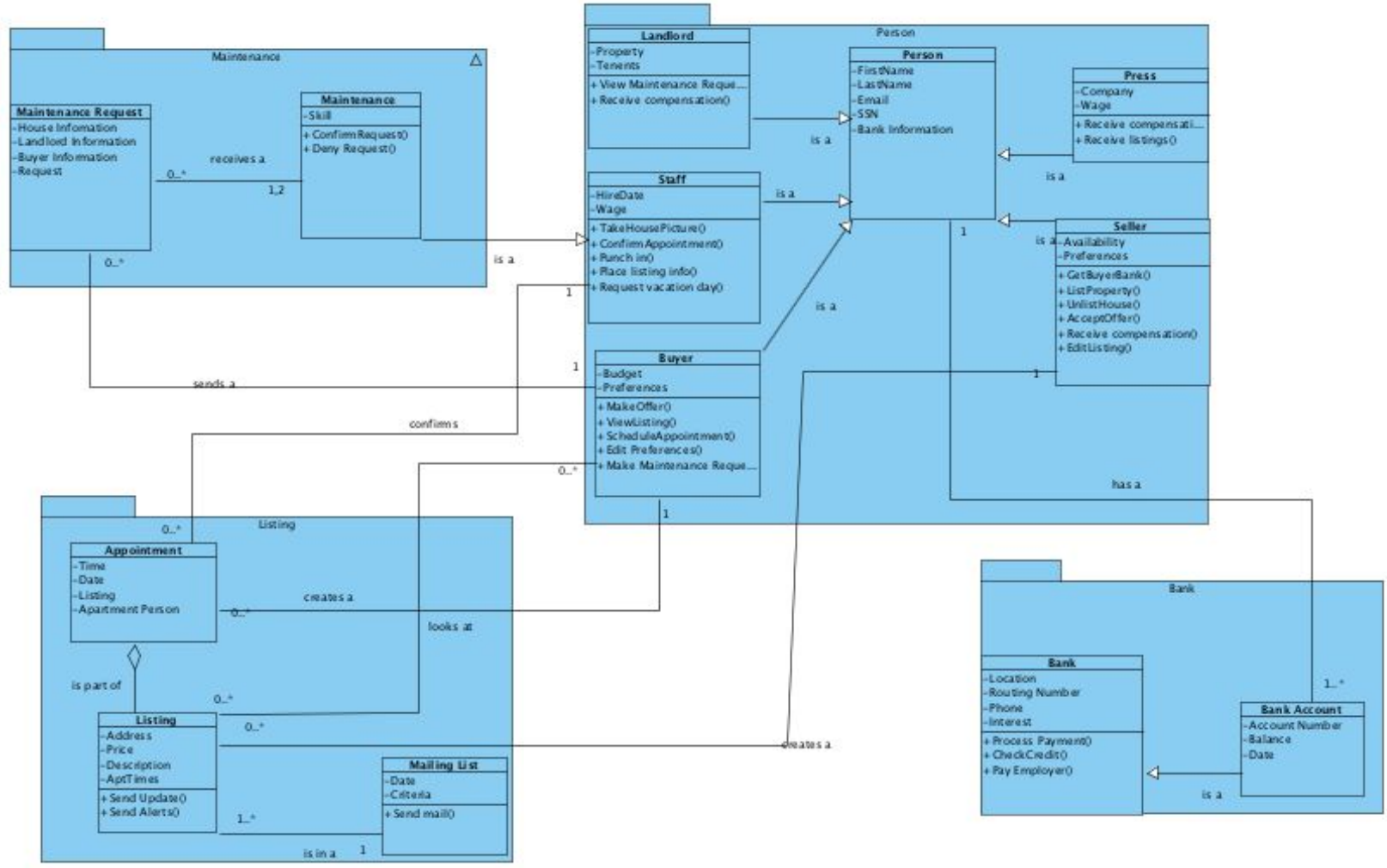
| | | |
|---|---|---|
| | 6) Staff selects print option | 5) Gives staff option to print request information<br><br>6) System send info to company printer |
| Alternate Courses: | 2a) No requests in system, returns no results<br>1a) Staff no longer wants to look at requests, clicks “Home” | |
| Conclusion: | Staff is able to see all requests made for maintenance | |
| Postcondition: | Request is confirmed | |
| Business Rules: | Only maintenance staff can see submitted requests | |
| Impl. Constraints and Specifications: | None | |
| Assumptions: | Assuming Staff prints when on a company computer NOT a laptop | |
| Open Issues: | None | |


View Maintenance Requests
Select maintenance request page
Requests load
Yes
No
Select Request from menu
Print and email request
Return no results

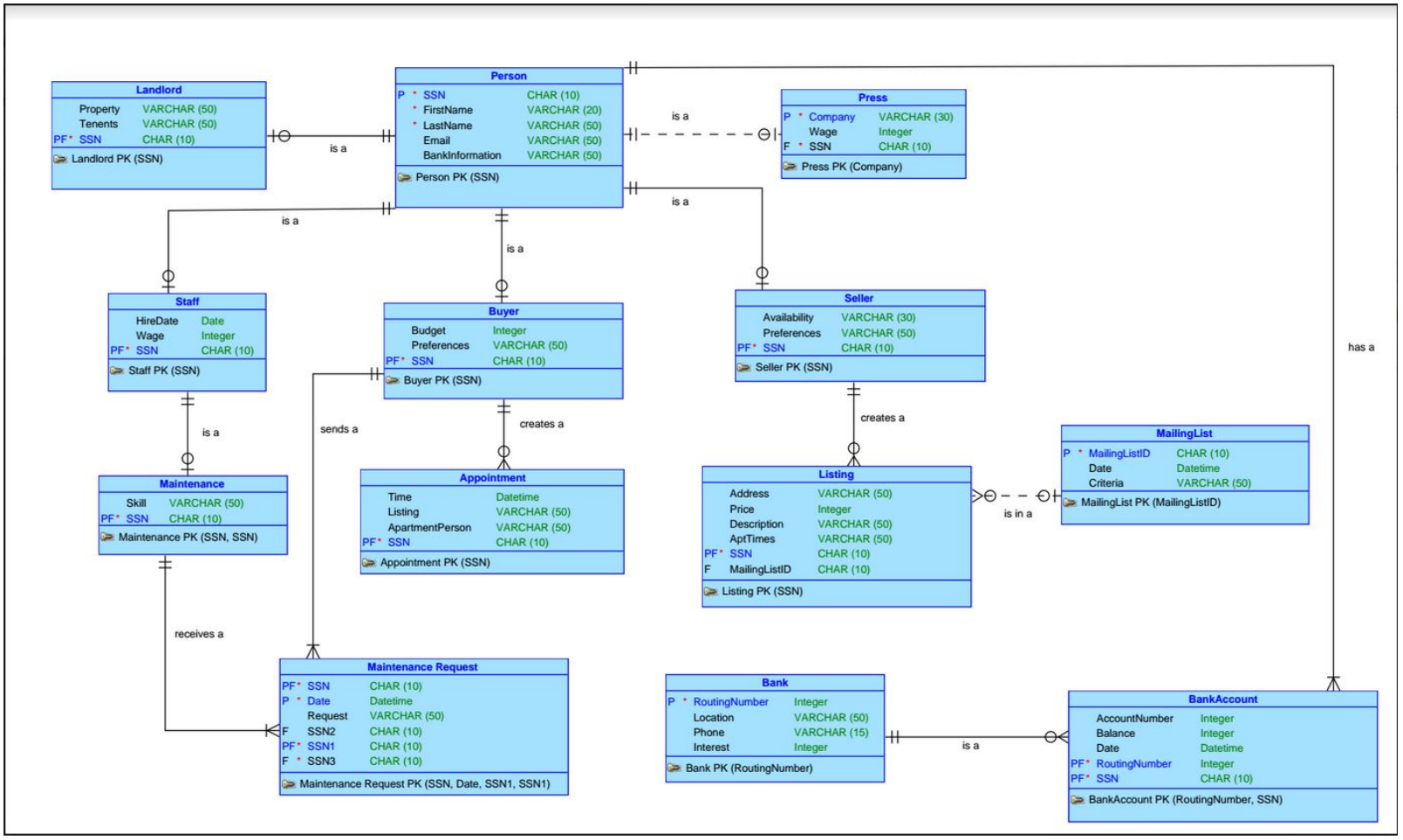

sd View Maintenance Requests
Staff
GUI
aRequest
aPrinter
1: Select Maintenance Request Page
1.1: Show Requests
2: Select Request
2.1: Give Option to Print
3: Select Print
3.1: Send to Company Printer
3.1.1: Confirm


sd View Maintenance Requests
Staff
GUI
aRequest
1: Select Maintenance Request Page
1.1: No Requests

## Class Diagram with Packages:



## Entity Relationship Diagram:

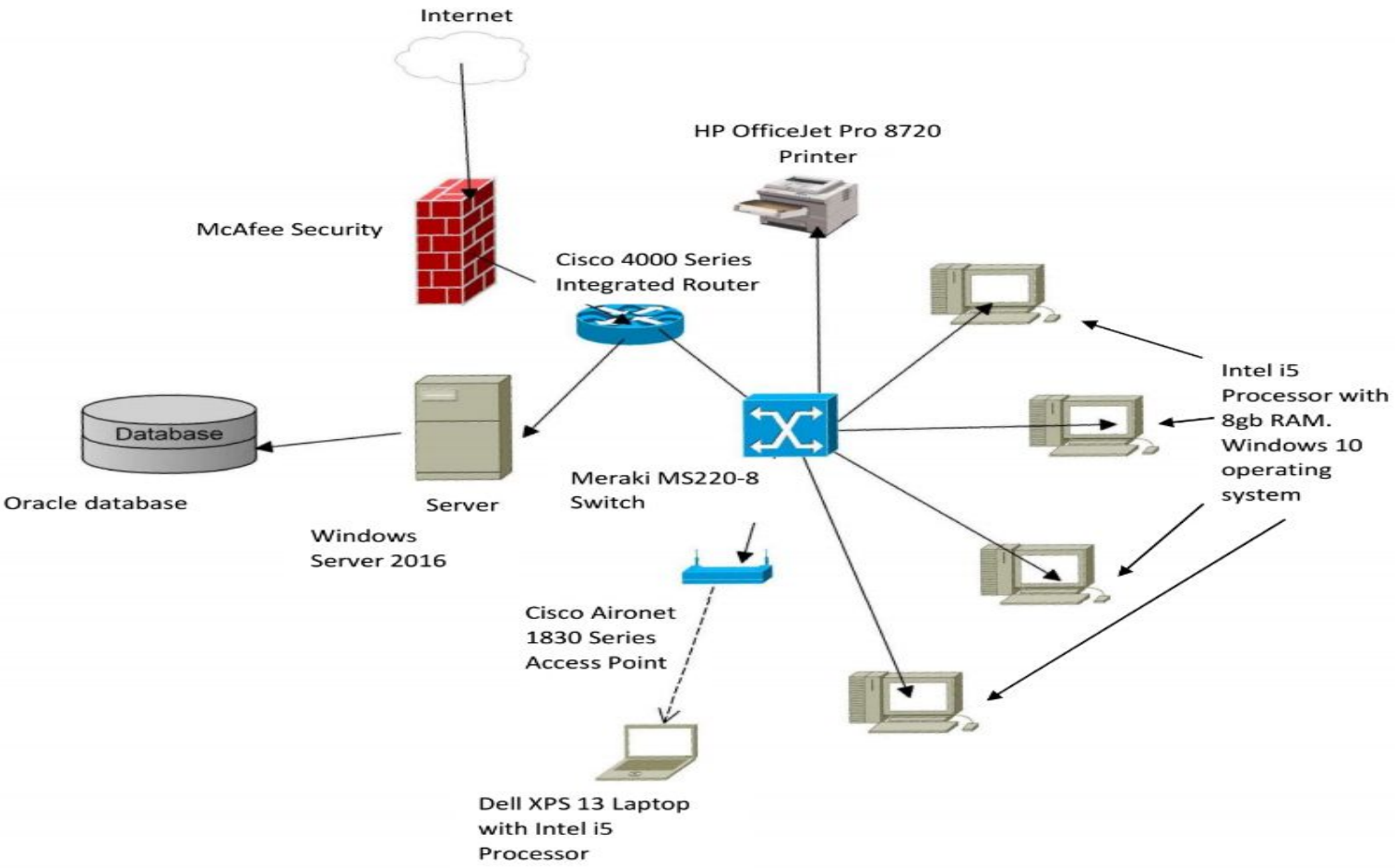


## Product Backlog

| Critical Order | ID | Item | Type | Status | Estimate (Weeks) |
|---|---|---|---|---|---|
| 11 | 001 | As a seller, I want to be able to receive offers from the buyers so that I know which buyer is the most interested. | Functional | Completed | 0.5 |
| 9 | 002 | As a seller of HSH, I want to be able to receive money so that I can receive the payment I deserve. | Functional | Completed | 0.5 |
| 1 | 003 | As a buyer, I want to be able to create a profile so that I am able see the houses that I am interested in. | Functional | Completed | 0.5 |
| 10 | 005 | As a buyer, I want the ability to withdraw offers from specific houses in order to be competitive in other housing bids. | Functional | Completed | 0.5 |
| 8 | 006 | As a buyer, I want the ability to make offers on specific houses in order to be competitive in housing bids. | Functional | Completed | 0.5 |
| 4 | 007 | As a landlord/maintenance, I want to give the ability for buyers to submit a maintenance request so that it can be easily accessed. | Functional | Completed | 0.5 |
| 3 | 010 | As a staff of HSH, I want to have a system that is able to hold 1,000+ records so that there can never be a limit to how many customers' account I can make | Non-functional performance | Completed | 0.5 |
| *14 | 012 | As a staff member of HSH, I want to be able to punch in and punch out so that my time spent working is accounted for, and I receive proper pay. | Functional | Completed | 0.5 |
| 7 | 013 | As a staff of HSH, I want to be able to take houses off the informational mailing list when they are no longer on the market. | Functional | Completed | 0.5 |
| 2 | 014 | As a seller, I want to be able to edit the information for my listing so that the most recent information is available. | Functional | Completed | 0.5 |
| 5 | 015 | As a buyer, I should be able to sort listing by price and location so that I can find the best listing. | Functional | Completed | 0.5 |
| 6 | 016 | As a buyer, I want to be able to make appointments so that I can visit the house | Functional | Completed | 0.5 |

| | | | | | |
|---|---|---|---|---|---|
| 13 | 017 | As a seller, I want to be able to keep a record of all of my sells so that I can have it incase of legal purposes | Functional | Completed | 0.5 |
| 15 | 018 | As a maintenance, I want to be able to see what maintenance requests there are available so that I can accomplish those requests | Functional | Completed | 0.5 |

## Deployment Diagram



## Cash Flow

| | 2017 | 2018 | 2019 | 2020 | 2021 | Total |
|---|---|---|---|---|---|---|
| Increased sales | 8,000 | 15,000 | 20,000 | 25,000 | 26,000 | |
| Reduction in customer complaint calls | 1,000 | 1,500 | 1,500 | 1,500 | 1,500 | |
| Reduced market time | 10,000 | 15,000 | 15,000 | 15,000 | 15,000 | |
| **TOTAL BENEFITS:** | **19,000** | **31,500** | **36,500** | **41,500** | **42,500** | |
| **PV OF BENEFITS:** | $ 18,447 | $ 29,692 | $ 33,403 | $ 36,872 | $ 36,661 | $ 155,074 |
| **PV OF ALL BENEFITS** | **$ 18,447** | **$ 48,138** | **$ 81,541** | **$ 118,413** | **$ 155,074** | |
| 1 Windows 2016 Server | 5,000 | 0 | 0 | 0 | 0 | |
| 4 Windows 10 Desktops | 2,000 | 0 | 0 | 0 | 0 | |
| 1 Cisco Aironet Router | 100 | 0 | 0 | 0 | 0 | |
| 1 Meraki switch | 100 | 0 | 0 | 0 | 0 | |
| 1 Dell XPS 13 laptop | 1,000 | 0 | 0 | 0 | 0 | |
| 1 OfficeJet Pro Printer | 1,000 | 0 | 0 | 0 | 0 | |
| Software licences | 5,000 | 0 | 0 | 0 | 0 | |
| Server software | 5,000 | 0 | 0 | 0 | 0 | |
| Development labor | 10,000 | 0 | 0 | 0 | 0 | |
| **TOTAL DEVELOPMENT COSTS:** | **29,200** | **0** | **0** | **0** | **0** | |
| Operational labor | 20,000 | 20,000 | 20,000 | 20,000 | 20,000 | |
| **TOTAL OPERATIONAL COSTS:** | **20,000** | **20,000** | **20,000** | **20,000** | **20,000** | |
| **TOTAL COSTS:** | **49,200** | **20,000** | **20,000** | **20,000** | **20,000** | |
| **PV OF COSTS** | $ 47,767 | $ 18,852 | $ 18,303 | $ 17,770 | $ 17,252 | $ 119,944 |
| **PV of ALL COSTS:** | **$ 47,767** | **$ 66,619** | **$ 84,922** | **$ 102,691** | **$ 119,944** | |
| **TOTAL PROJECT BENEFITS-COSTS:** | **$ (30,200)** | **$ 11,500** | **$ 16,500** | **$ 21,500** | **$ 22,500** | |
| **YEARLY NPV:** | **$ (29,320)** | **$ 10,840** | **$ 15,100** | **$ 19,102** | **$ 19,409** | |
| **CUMULATIVE NPV:** | **$ (29,320)** | **$ (18,481)** | **$ (3,381)** | **$ 15,722** | **$ 35,130** | |
| **RETURN ON INVESTMENT:** | **29.29%** | | | | | |
| **BREAK-EVEN POINT:** | **3.18** | | | | | |

# Gantt Chart

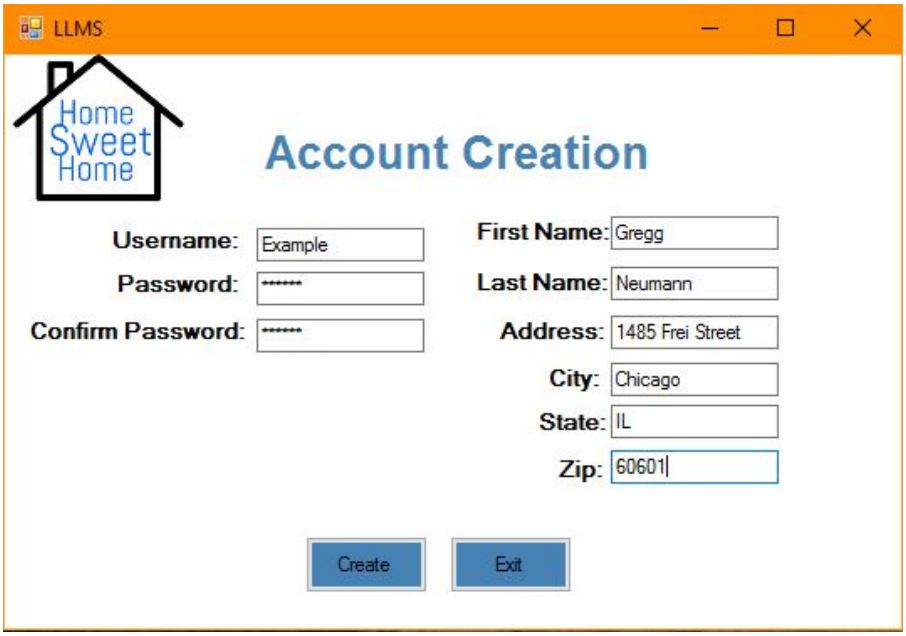
| Project | Home Sweet Home Business System | | Week 1 | Week 2 | Week 3 | Week 4 | Week 5 | Week 6 | Week 7 | Week 8 | Week 9 | Week 10 | Week 11 | Week 12 | Week 13 | Week 14 | Week 15 | Week 16 | Week 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Starte Date | 08/25/2016 | Available Weeks | 0 | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done | Done |
| Due Date | 12/08/2016 | Percentage Completed | 100.00% | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |
| Task Order | Task ID | Task Description | Percentage done | | | | | | | | | | | | | | | | | |
| 1 | 3 | Buyer creates profile | 100% | | | | | | | | | | | | | | | | | |
| 2 | 14 | Seller edits listing information | 100% | | | | | | | | | | | | | | | | | |
| 3 | 10 | System able to hold many reconds | 100% | | | | | | | | | | | | | | | | | |
| 4 | 7 | Buyer submits maintenance request | 100% | | | | | | | | | | | | | | | | | |
| 5 | 15 | Buyer sorts listings | 100% | | | | | | | | | | | | | | | | | |
| 6 | 16 | Buyer makes appointments | 100% | | | | | | | | | | | | | | | | | |
| 7 | 13 | Seller takes down listings | 100% | | | | | | | | | | | | | | | | | |
| 8 | 6 | Buyer makes offer | 100% | | | | | | | | | | | | | | | | | |
| 9 | 2 | Seller received money | 100% | | | | | | | | | | | | | | | | | |
| 10 | 5 | Buyer withdraws offers | 100% | | | | | | | | | | | | | | | | | |
| 11 | 1 | Seller receives offers | 100% | | | | | | | | | | | | | | | | | |
| 13 | 17 | Seller views their records | 100% | | | | | | | | | | | | | | | | | |
| 14 | 12 | Staff enters work time | 100% | | | | | | | | | | | | | | | | | |
| 15 | 18 | Maintenance views requests | 100% | | | | | | | | | | | | | | | | | |
| 16 | 11 | Physical Design | 100% | | | | | | | | | | | | | | | | | |
| 12 | 9 | Cash Flow | 100% | | | | | | | | | | | | | | | | | |
| | | | | Mathew | Joshua | Benya | Kyle | Joey | | | | | | | | | | | | |

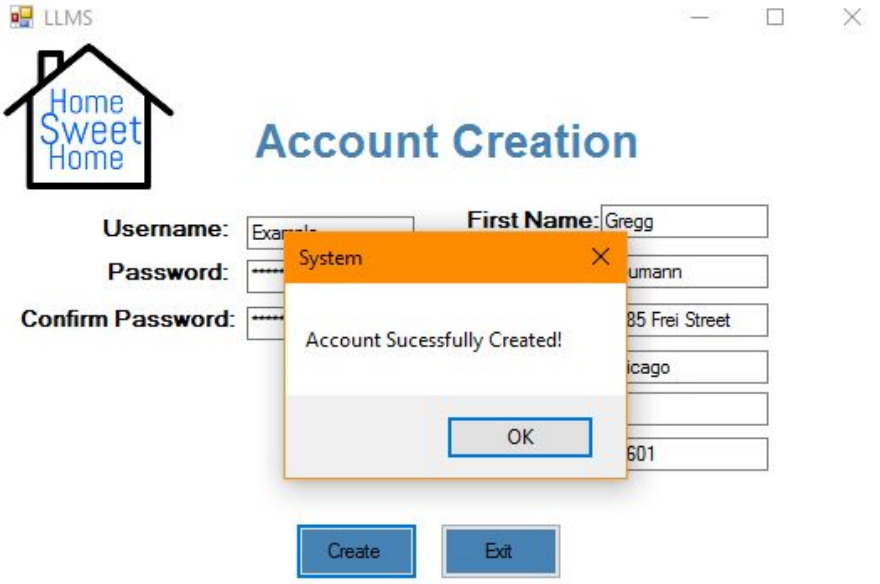
## Screenshots of the Home Sweet Home Business System by LLMS

This is the start up screen for Home Sweet Home Business System:


Create new Account page:

Fill out the form and select create:

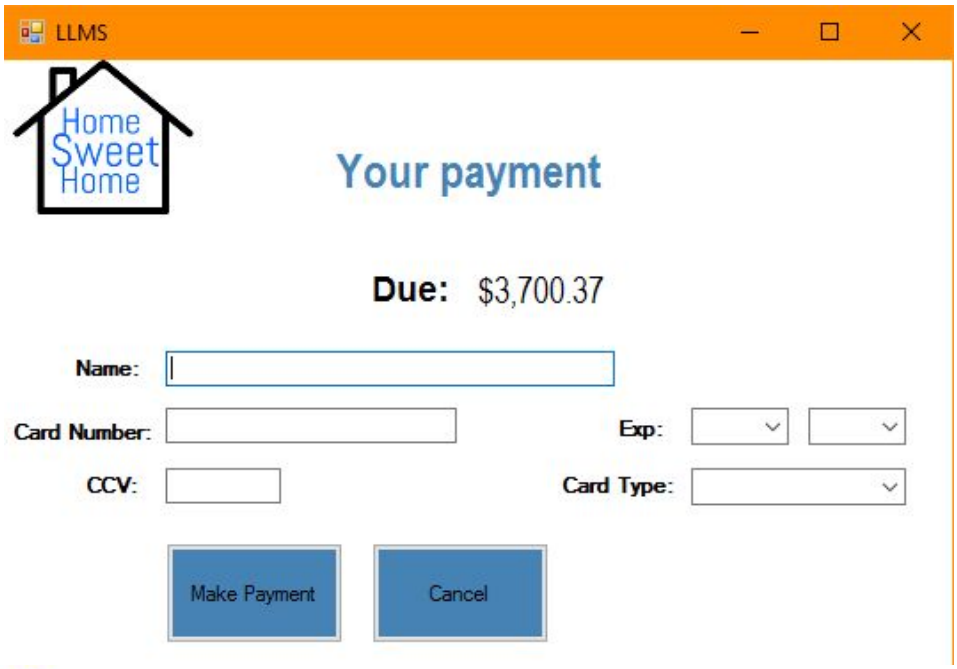


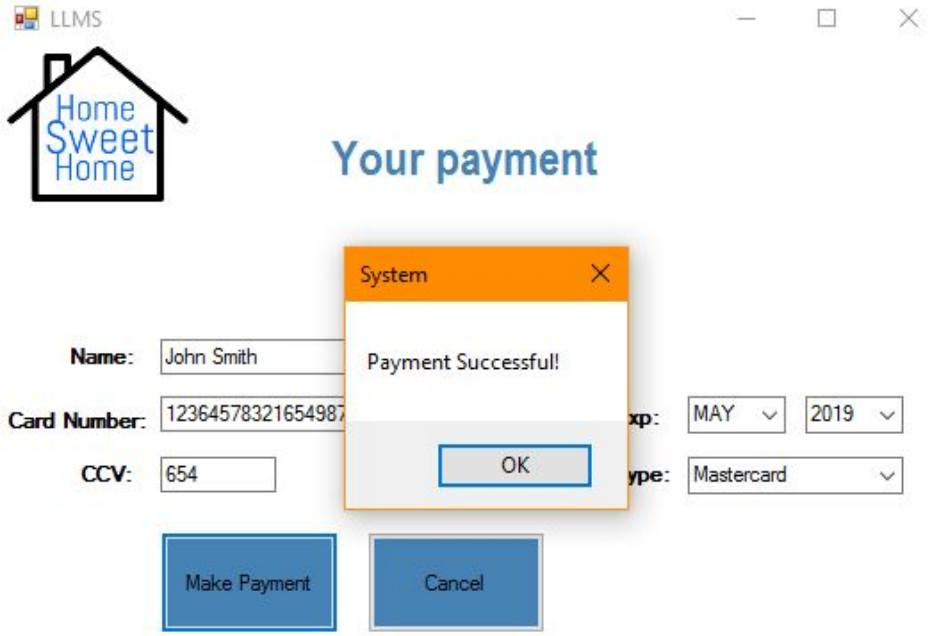



If we login as “Buyer”:

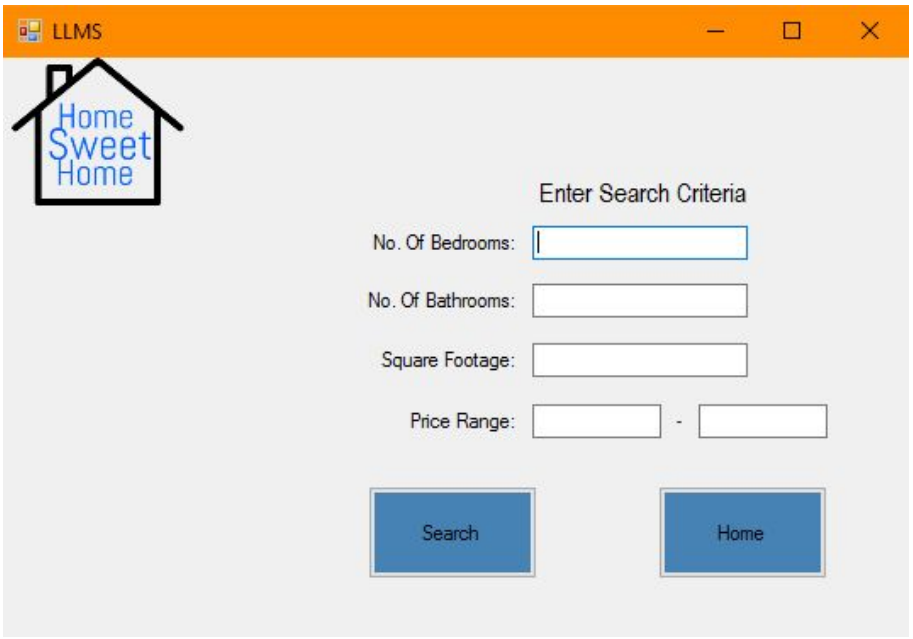

The Buyer homepage. All buyer actions start here:

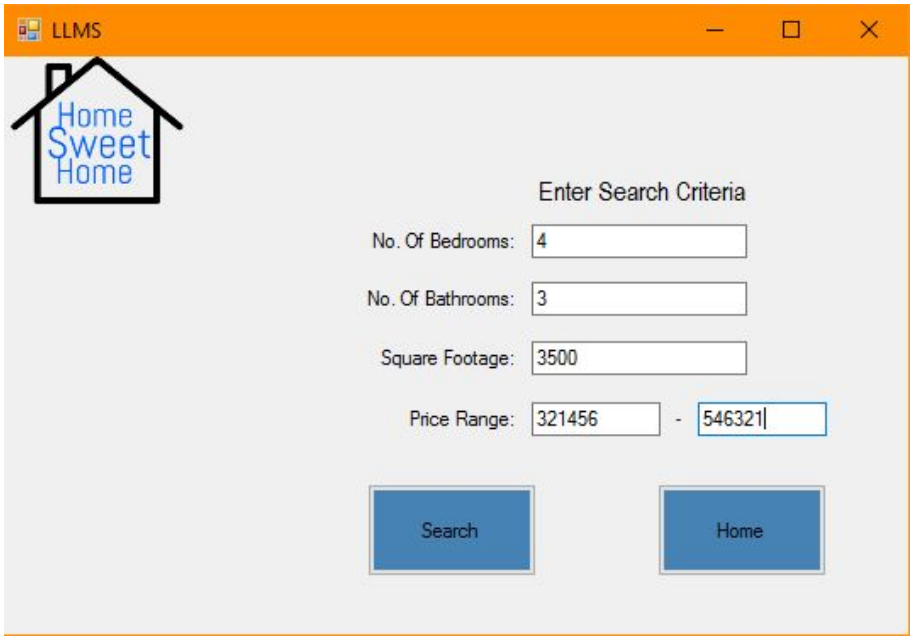
Select the Make Payment button:

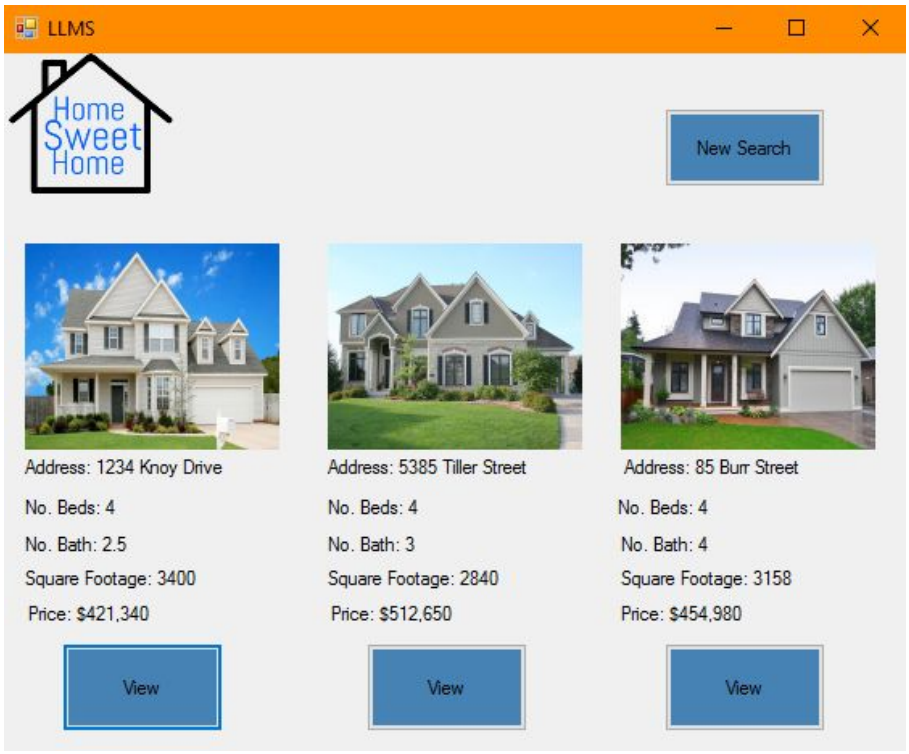


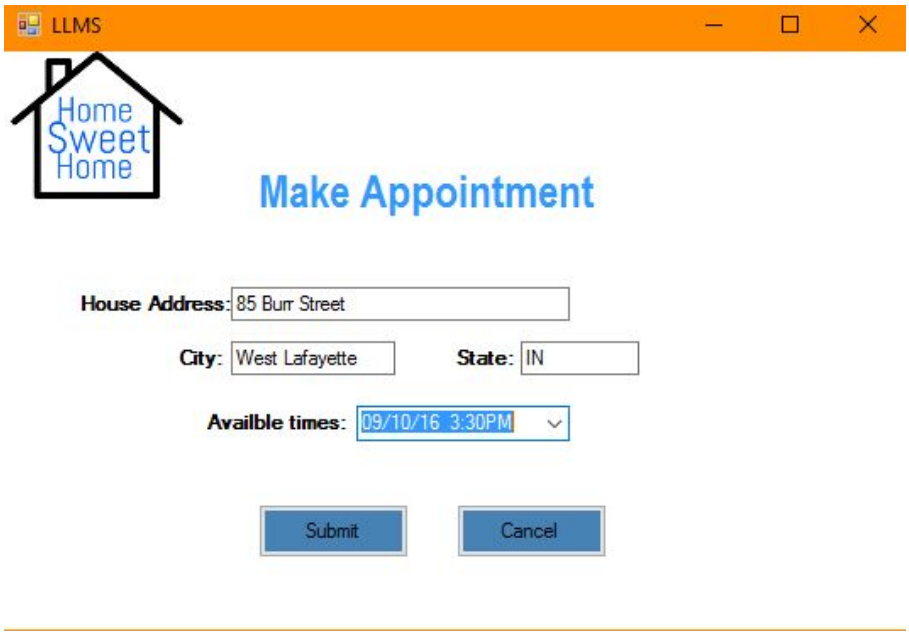



After returning the the homepage, select Search Listings then enter search criteria. The user can then select view to set up an appointment to view the home.:

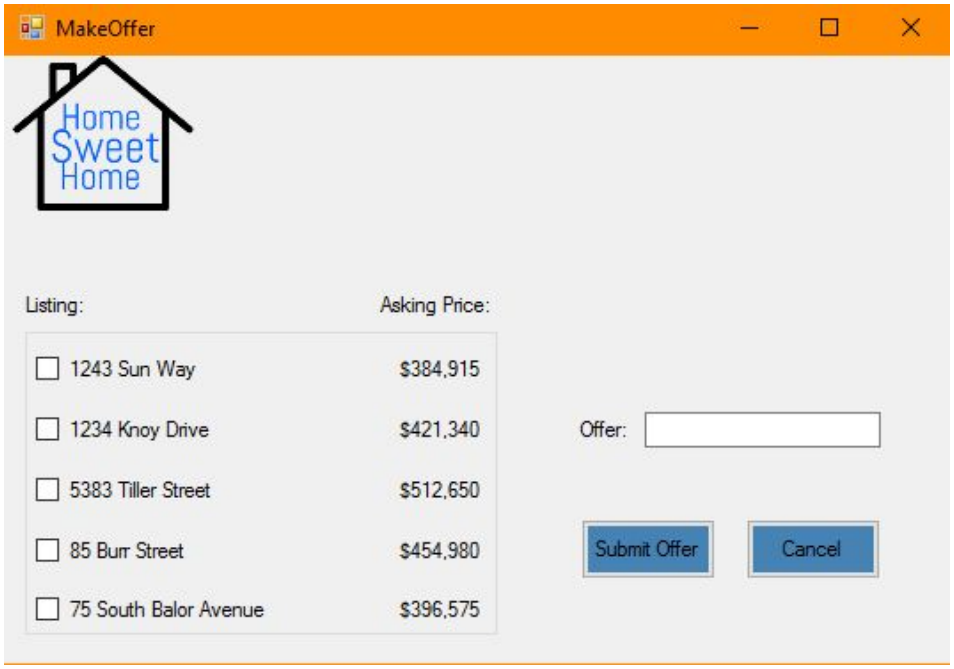


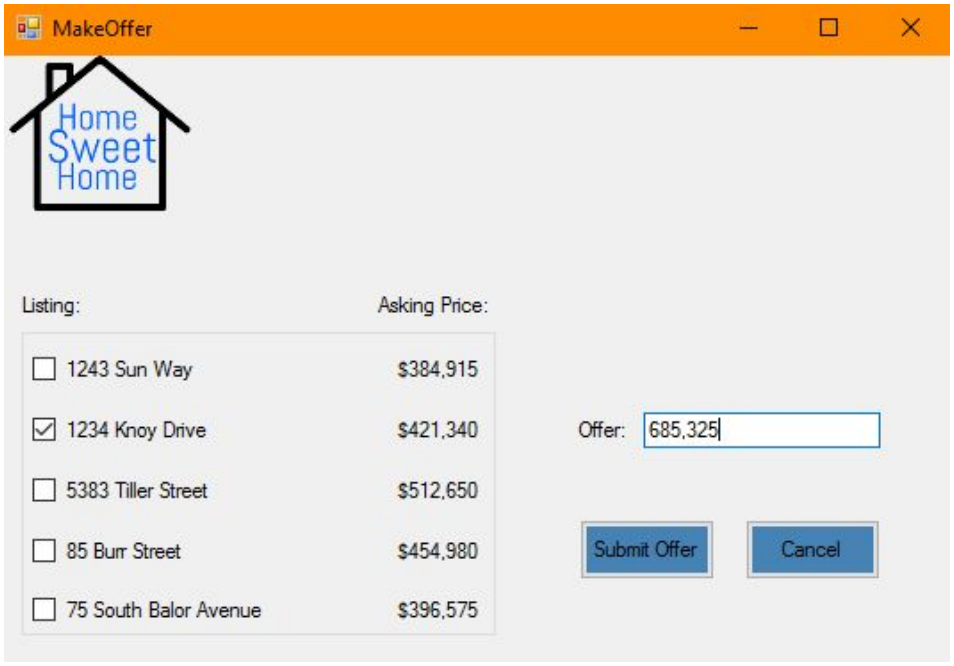



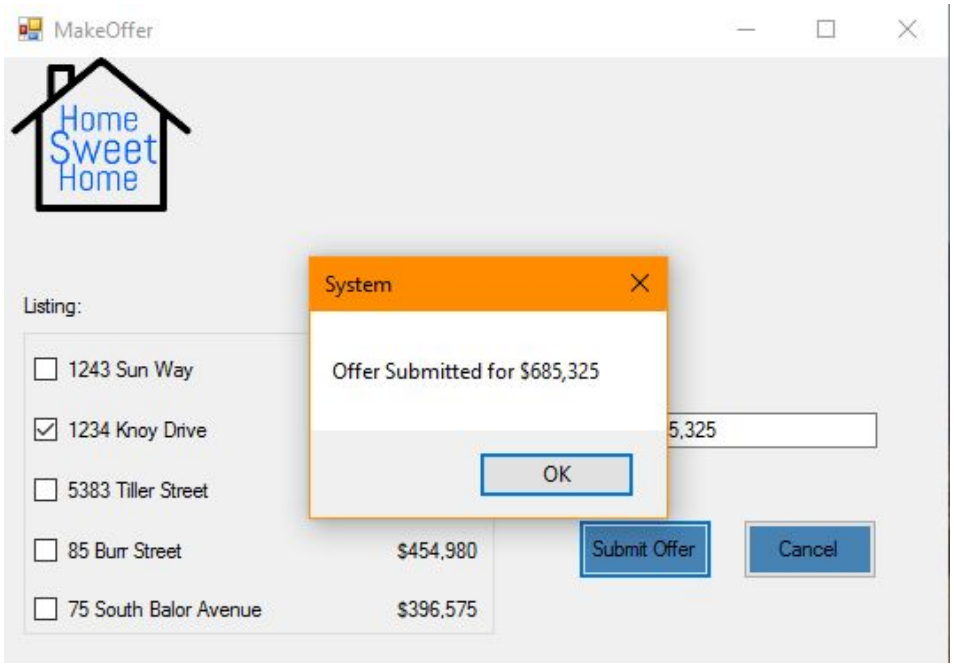

LLMS
Home Sweet Home
New Search
Address: 1234 Knoy Drive
No. Beds: 4
No. Bath: 2.5
Square Footage: 3400
Price: $421,340
View
Address: 5385 Tiller Street
No. Beds: 4
No. Bath: 3
Square Footage: 2840
Price: $512,650
View
Address: 85 Burr Street
No. Beds: 4
No. Bath: 4
Square Footage: 3158
Price: $454,980
View


LLMS
Home Sweet Home
Make Appointment
House Address: 85 Burr Street
City: West Lafayette
State: IN
Availble times: 09/10/16 3:30PM
Submit
Cancel

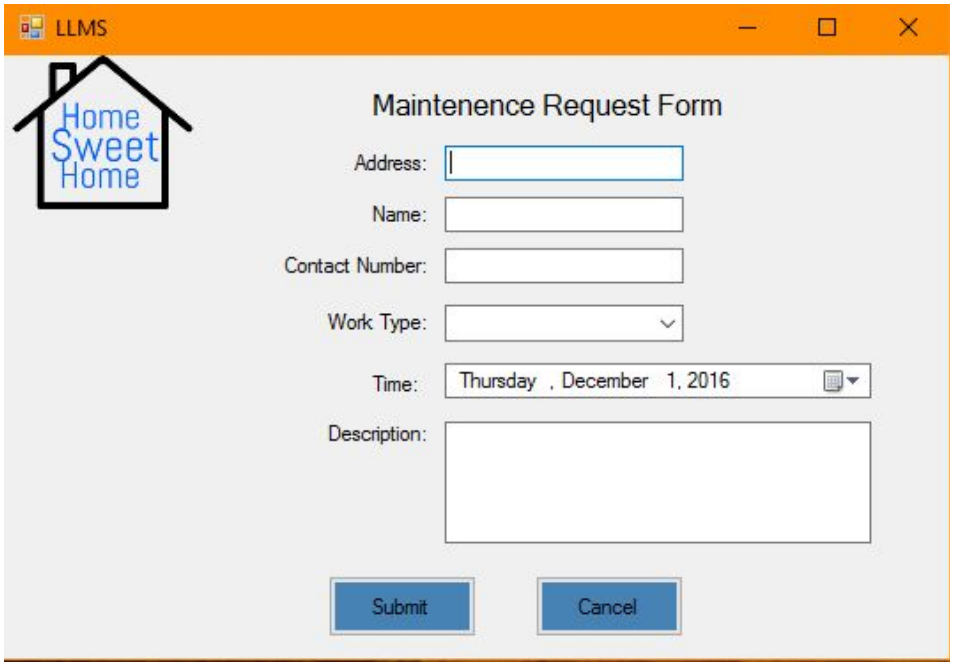
After returning to the homepage select the Make Offer button to make an offer on a home:

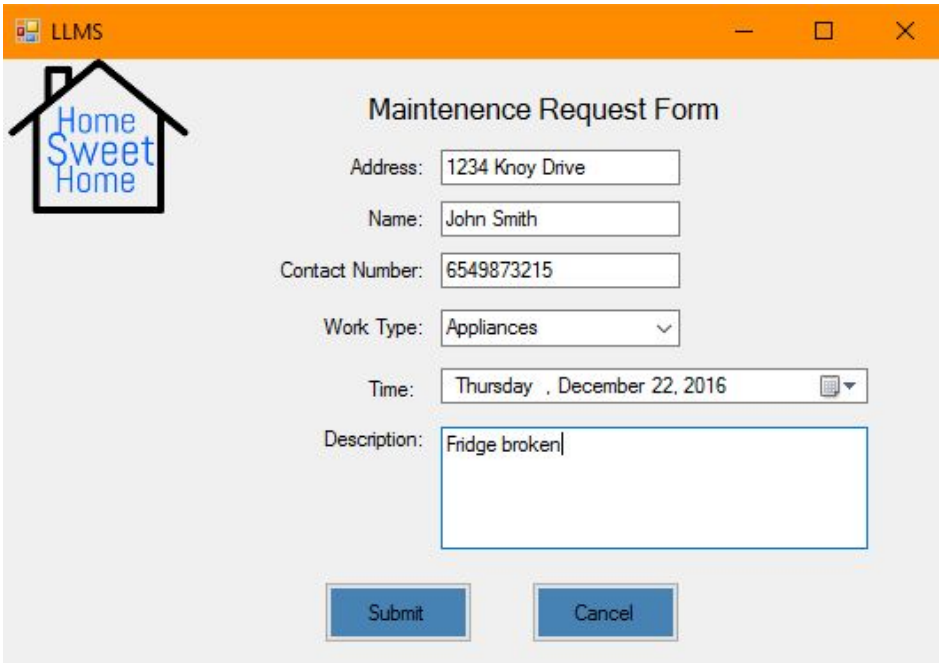


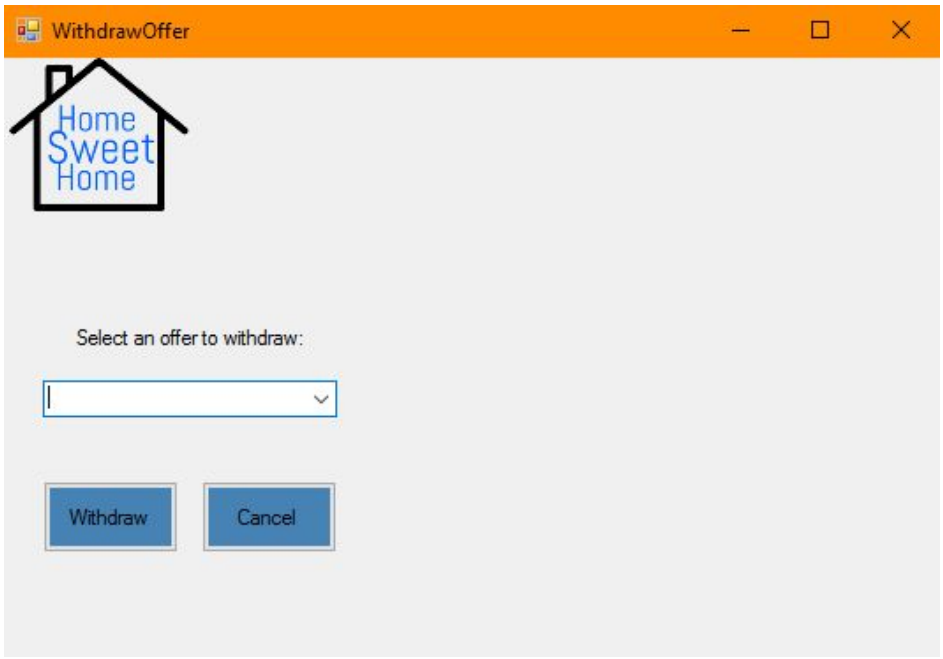


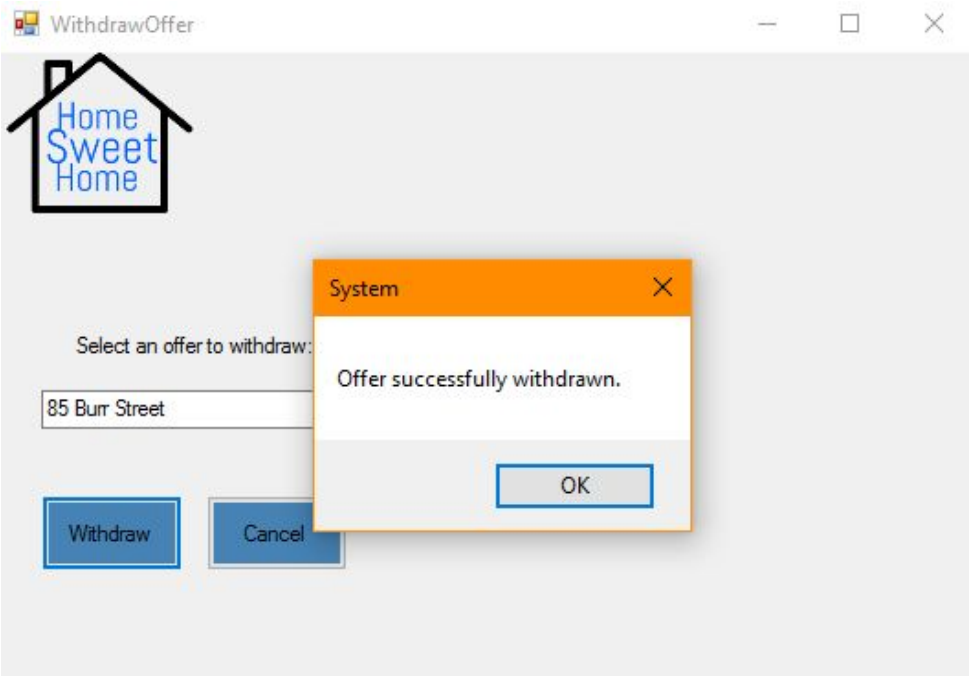

Return to the homepage and select the Maintenance Request button to fill out a work order:



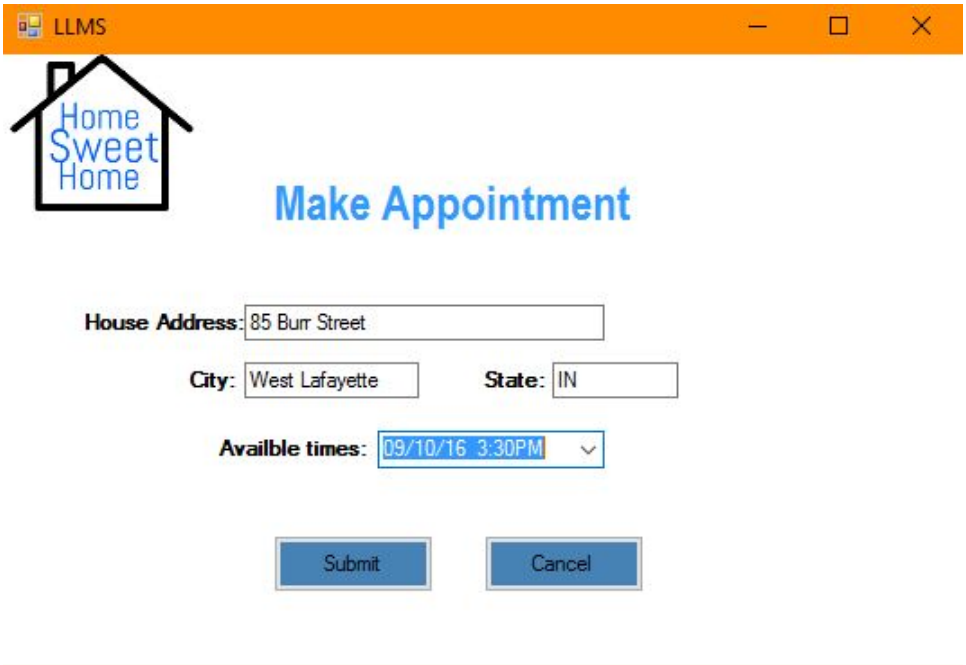



Return to the homepage and select the Withdraw Offer button to withdraw and offer:

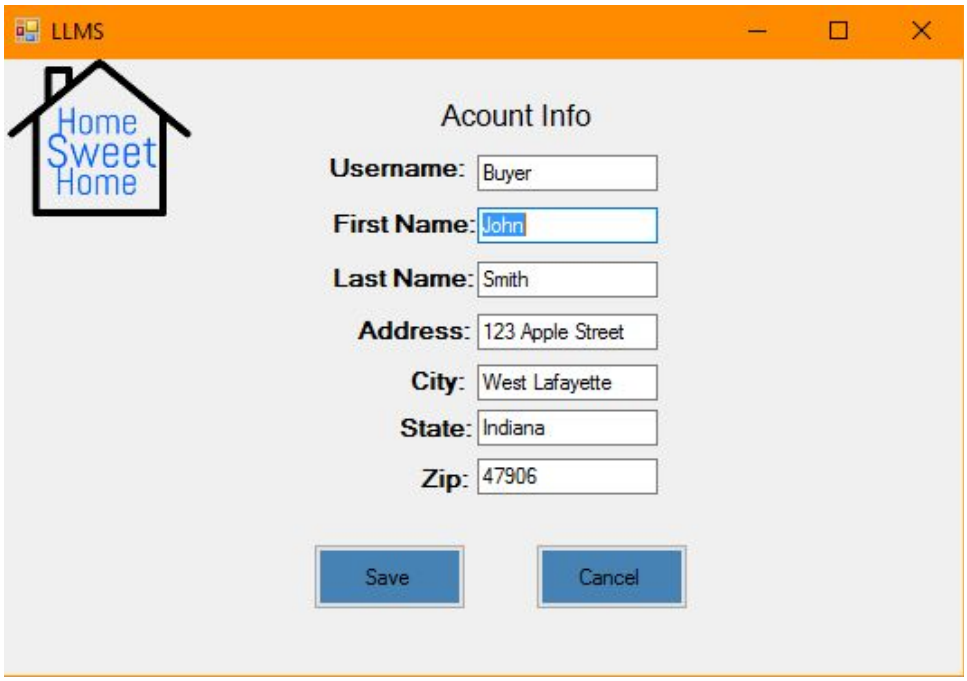




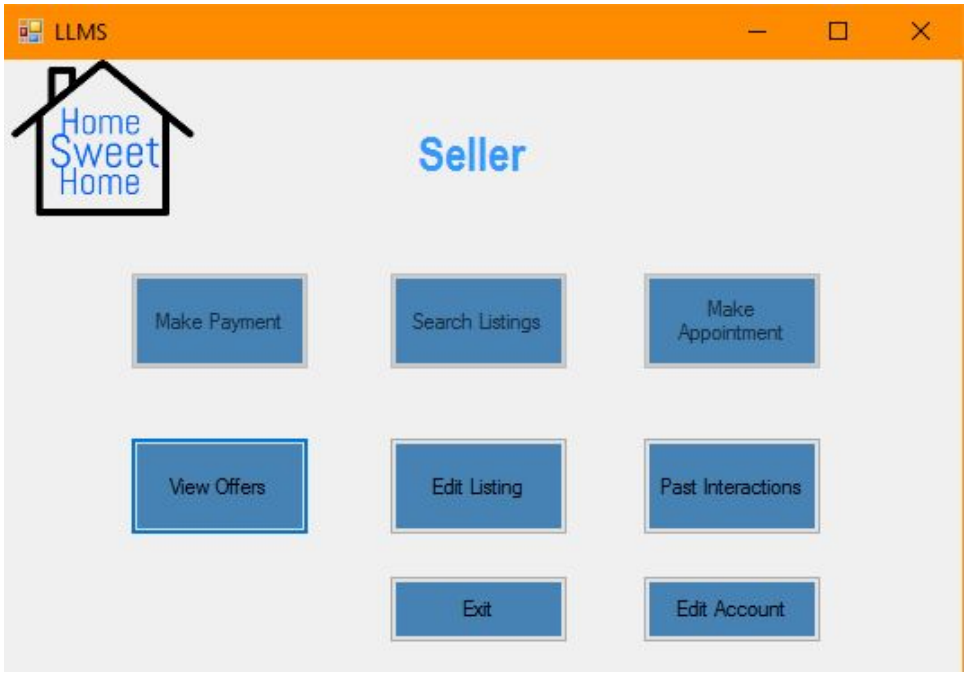

Return to the homepage and select the Make Appointment button to make and appointment to view a home:



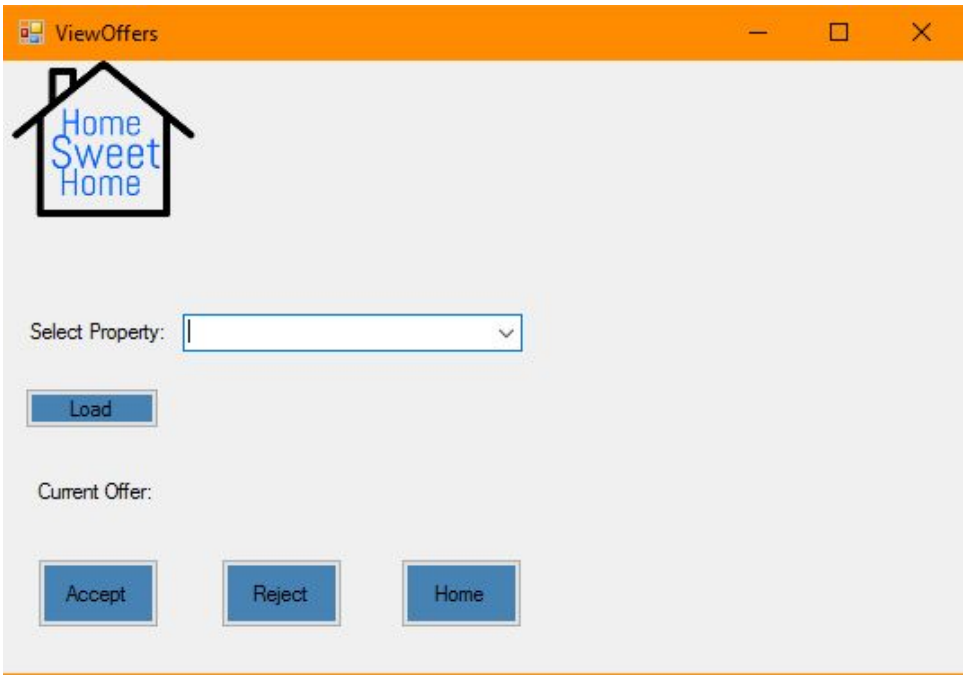
Return to the homepage and select the Edit Account button to edit account information:

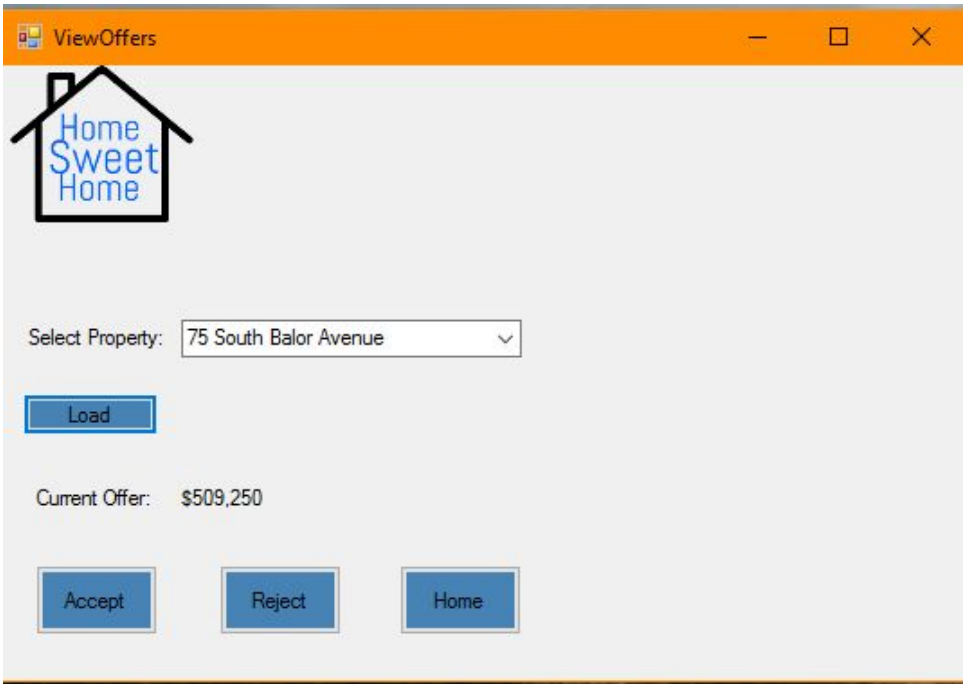



Next, restart the program to login as a Seller:

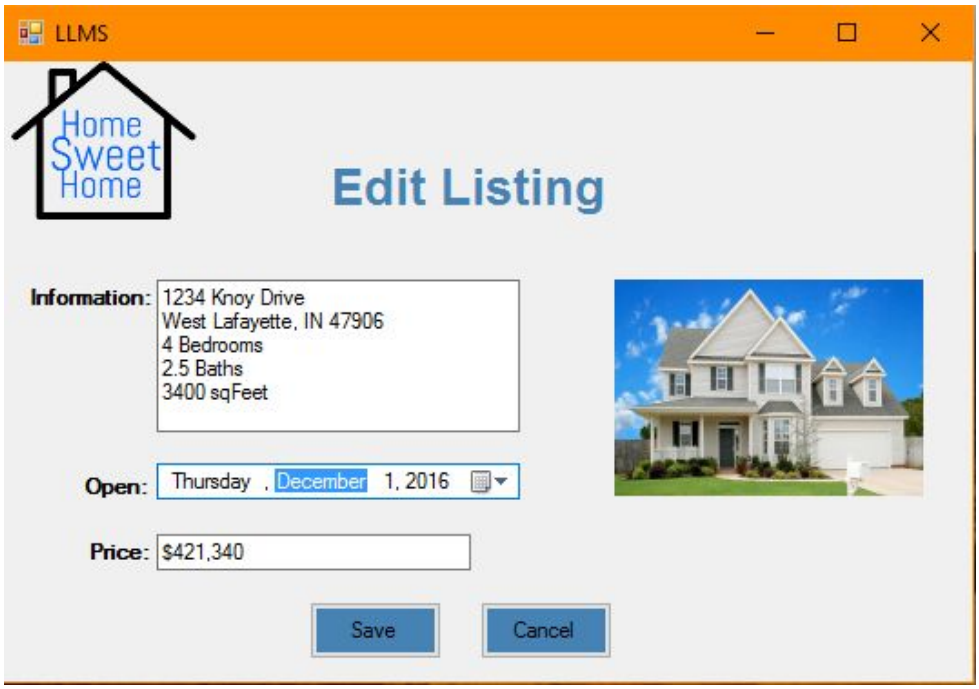


Select the View Offers button to view the offers made on the user's listings. The seller can then accept or reject the offer:

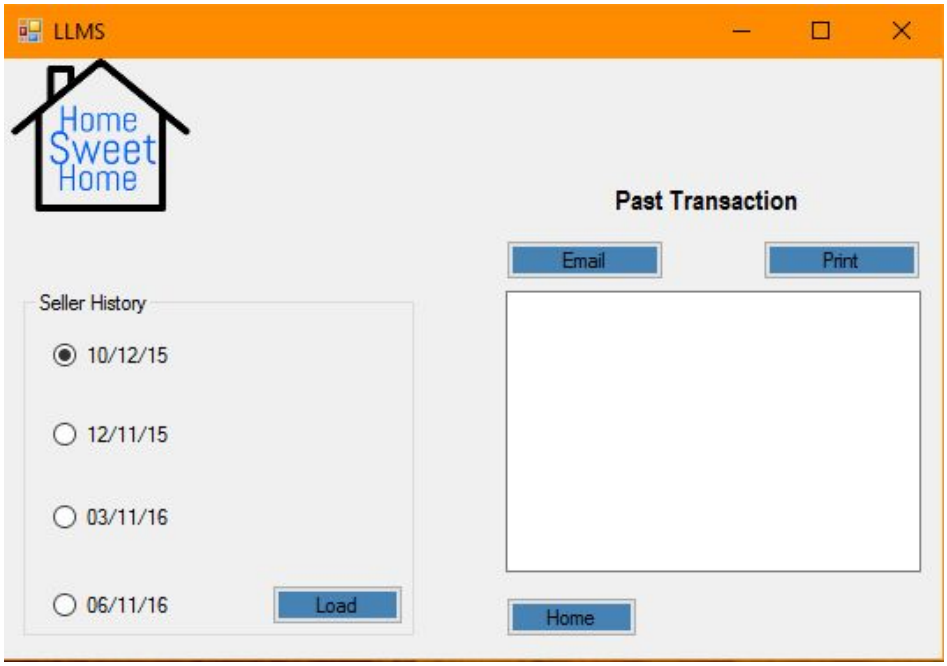


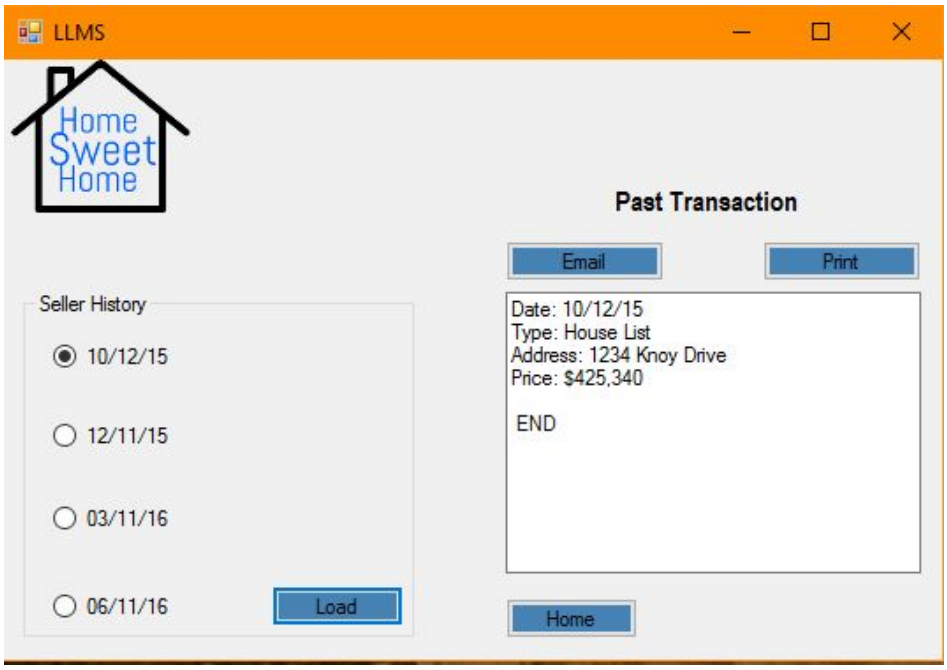



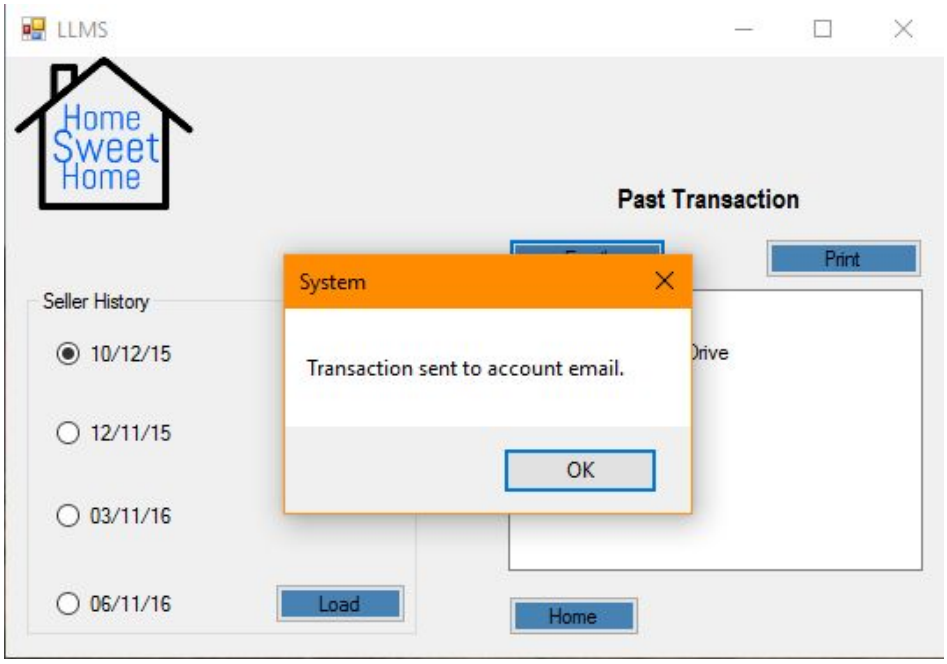
Return to the homepage and select the Edit Listings button to edit the listings:




Return to the homepage and select the Past Interactions button to view all past interactions with HSH. They can also email or print these interactions:

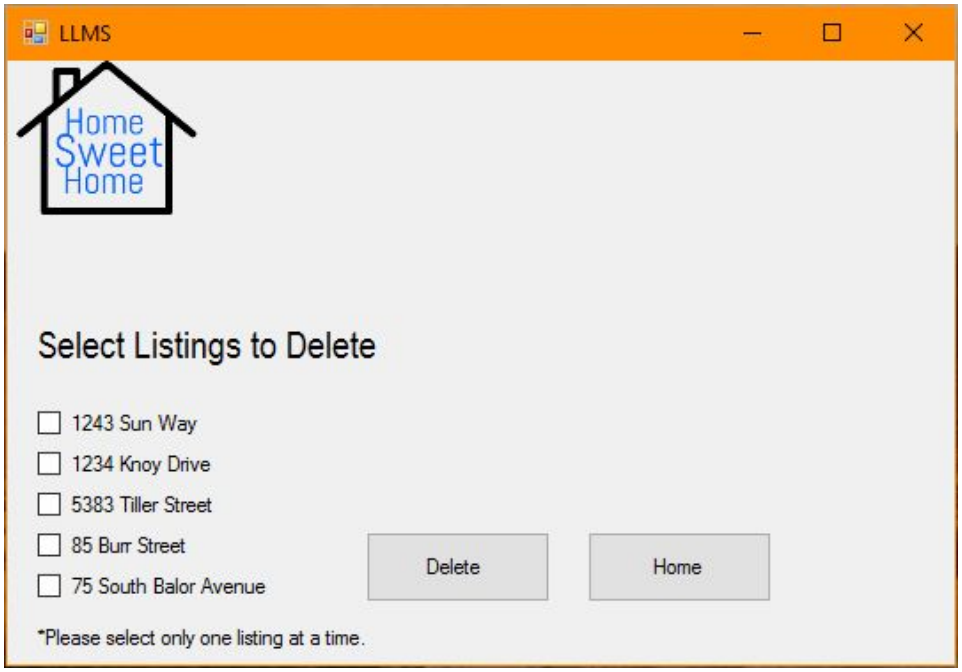


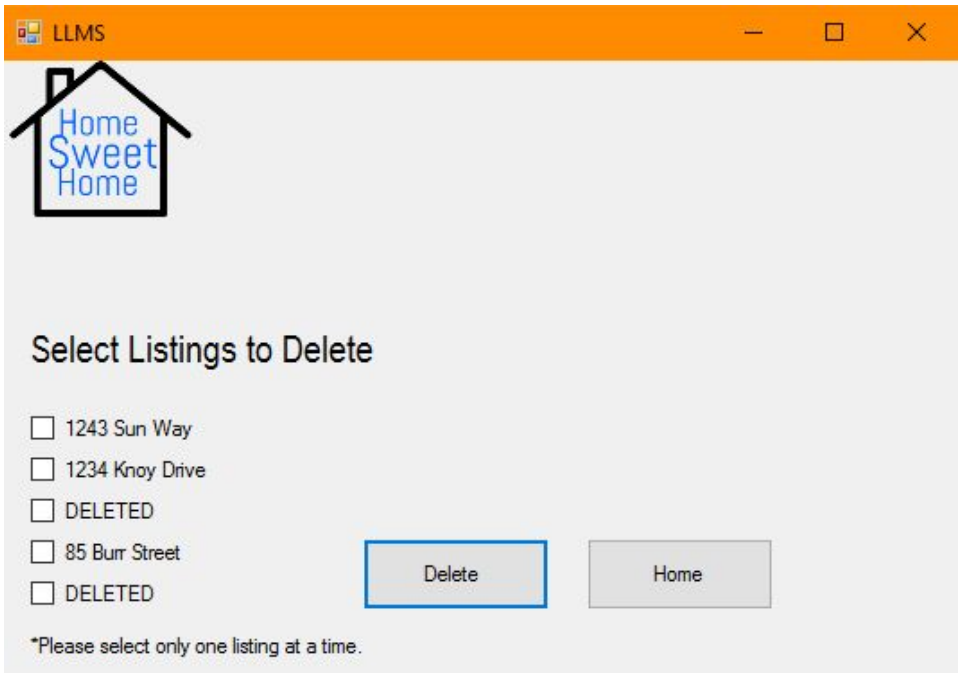

LLMS
Home Sweet Home
Past Transaction
Email
Print
Seller History
10/12/15
12/11/15
03/11/16
06/11/16
Load
Date: 10/12/15
Type: House List
Address: 1234 Knoy Drive
Price: $425,340
END
Home


LLMS
Home Sweet Home
Past Transaction
Print
Seller History
10/12/15
12/11/15
03/11/16
06/11/16
Load
Home
System
Transaction sent to account email.
OK

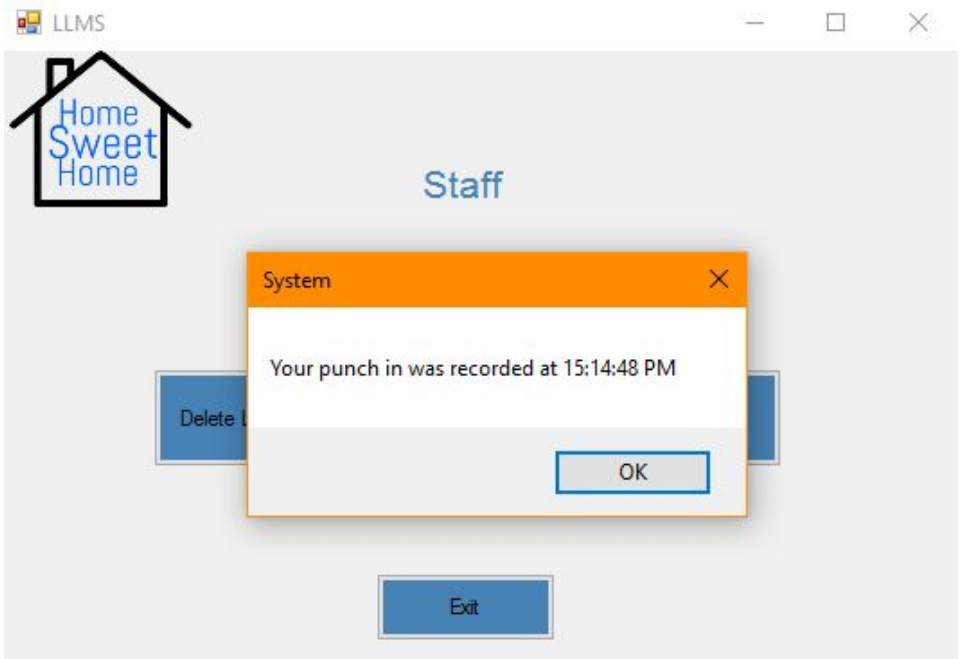
Next, restart the program to login as a Staff:


Select the Delete Listings button to delete listings from HSH:

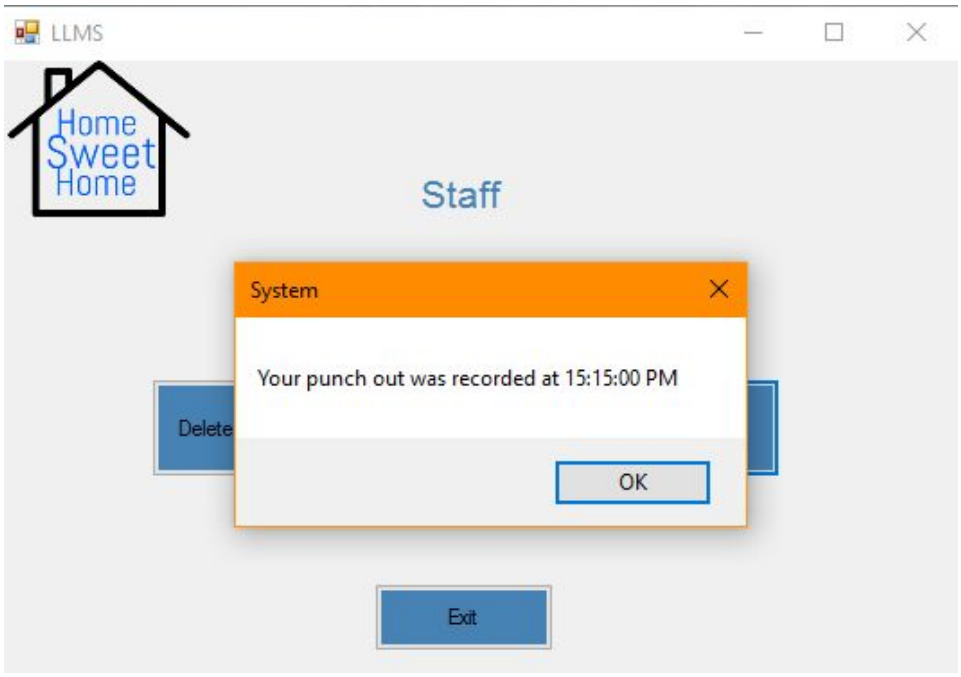




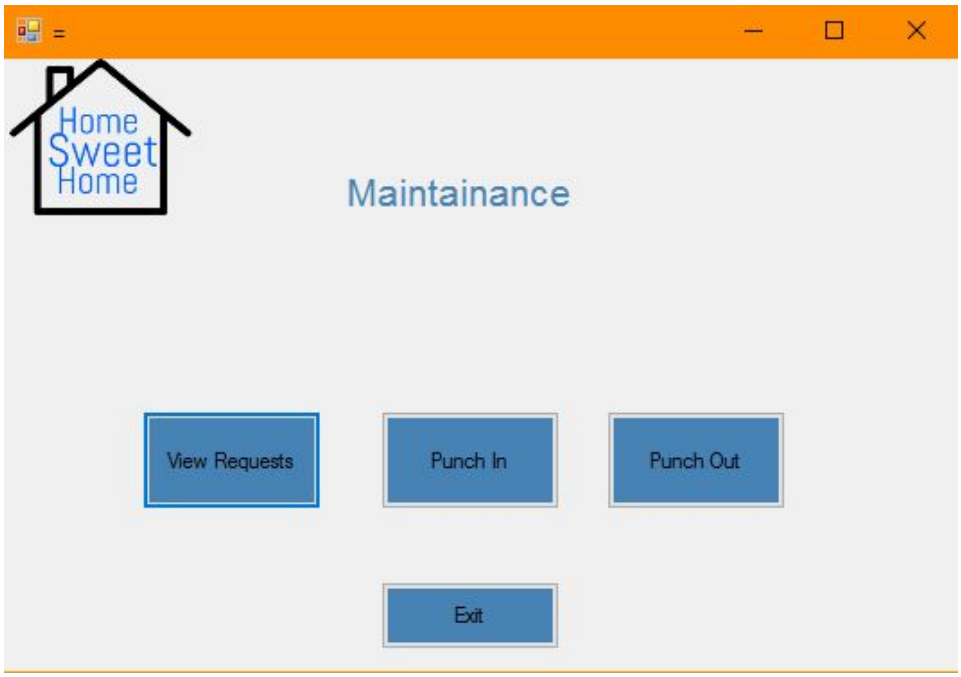

Return to the homepage and select the Punch In button to punch in:



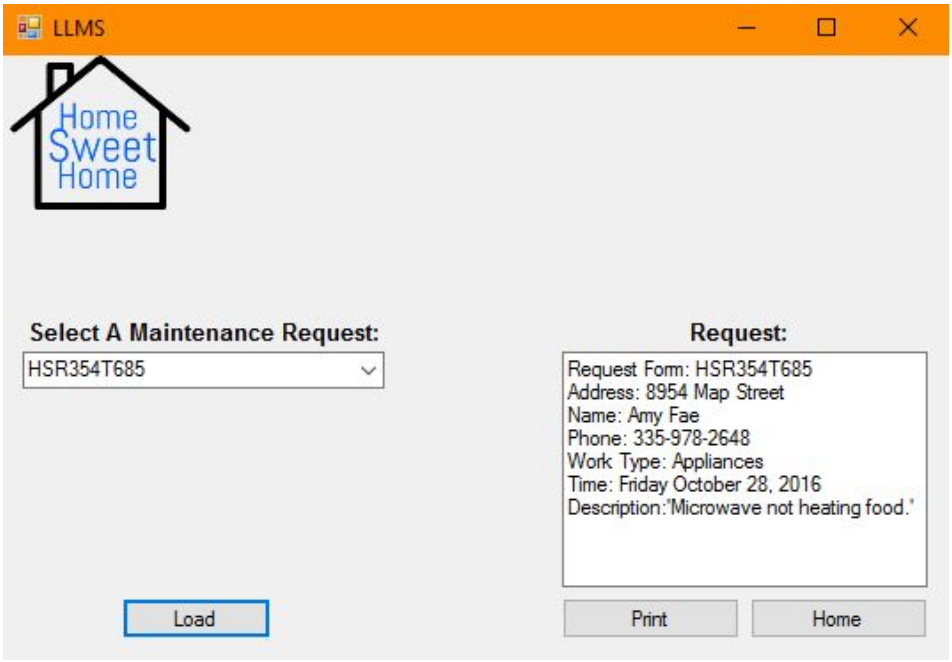
Return to the homepage and select the Punch Out button to punch out:

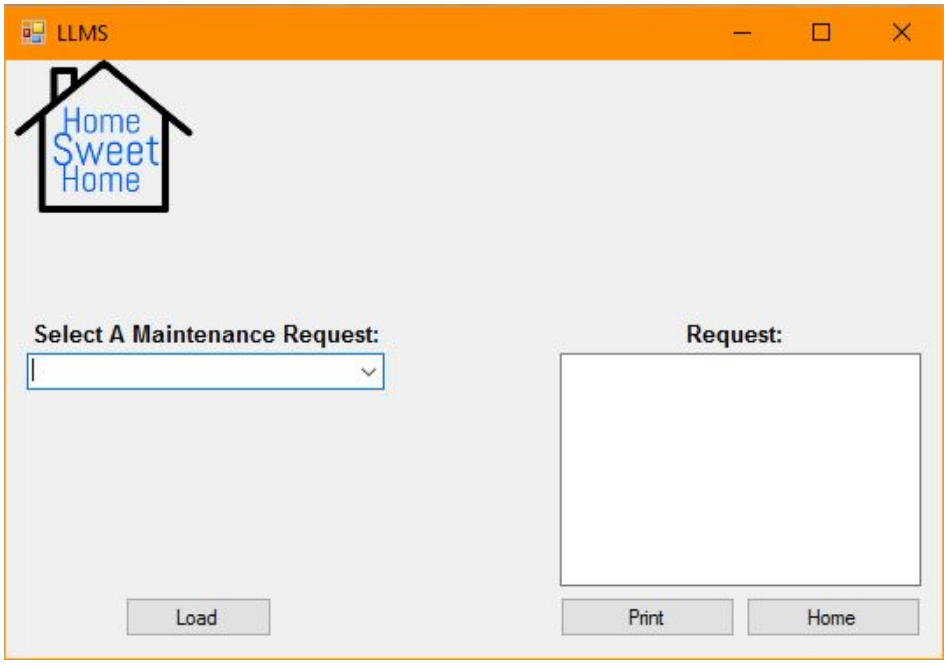



Restart the program and logon as Maint (Maintenance):



Select the View Requests button to view all of the maintenance requests in the system. The worker can then print them from this form. The maintenance worker also has the same ability to punch in and out.:

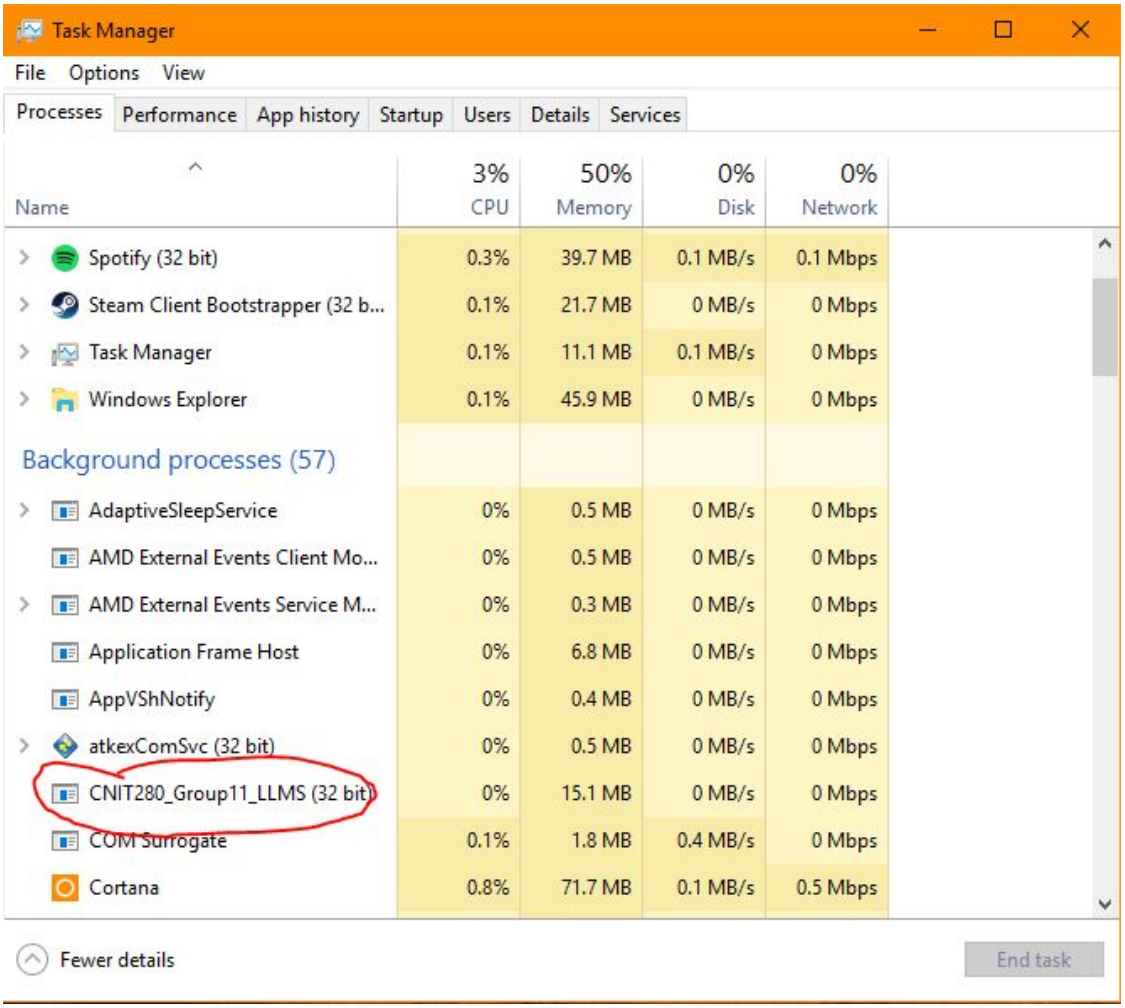


## Final product usernames and passwords

| Username | Password |
|---|---|
| Buyer | Buyer |
| Seller | Seller |
| Staff | Staff |
| Maint | Maint |

****NOTE**** Due to an error with form opening and closing in C# the program does not fully close when you exit. In order to fully shutdown the program, you must go to the task manager and find it under processes. You end the task from task manager.



## Preliminary Usability Testing:

Under each question of each dimension we have put an average score compiled from the individual scores of each tester. We tested our prototype with six different users including all five group members, and one outside user.

**Accessibility**

**1.** Color alone is not used to convey information.

**10**

**2.** Content is readable without a style sheet. Check e.g: Web Accessibility Checker

**9.5**

**3.** Accessible navigation. Site can be navigated with a keyboard, without using a mouse. Browser's keyboard shortcuts aren't overridden.

**7.5**

**4.** Links, buttons and checkboxes are easily clickable, for example a user can select a checkbox by clicking the text, not just the checkbox.

**8.8**

**Navigation**

**1.** Users know where they are on the site. For example, with the use of breadcrumbs. Also, there is a site map on large sites.

**8.3**

**2.** Navigation is consistent on every page.

9.5

3. Links are descriptive. There are no "click here" links.

**9.7**

**4.** Important links aren't placed in moving features**, for example auto-rotating carousels and accordions.**

**10**

**Layout**

1. Important content is displayed first.

9.5

2. Site is responsive. Works with different screen sizes. There is no horizontal scrolling.

**9.0**

**3.** Related information is grouped together clearly.

8.5

4. There are a minimum amount of pop-up windows.

8.2

5. Consistency. Page layouts are consistent across the whole website.

**9.2**

**Content**

**1.** Content is scannable. There are short paragraphs, descriptive headings, lists and images. Visual content is used when appropriate, instead of large amounts of text.

**9.5**

**2.** Content is written with common language that users easily understand. Check with e.g.: The Readability Test Tool.

**9.8**

**3.** Content is useful and up-to-date, providing answers to the most common questions asked by users. There are no long instructions or "welcome to our website" text.

**9.0**

**4.** Use of uppercase letters in prose text is avoided. Uppercase is used only for formatting.

**9.0**

Below is a picture of the spreadsheet used to calculate the averages and an example of one of the functions.

| Question | Benya | Joey | Kyle | Mathew | Mattew | Josh | Average |
|---|---|---|---|---|---|---|---|
| 1.1 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 1.2 | 9 | 10 | 10 | 10 | 9 | 9 | 9.5 |
| 1.3 | 7 | 7 | 8 | 7 | 8 | 8 | 7.5 |
| 1.4 | 8 | 9 | 10 | 7 | 9 | 10 | 8.8 |
| 2.1 | 9 | 8 | 8 | 8 | 7 | 10 | 8.3 |
| 2.2 | 10 | 10 | 10 | 10 | 7 | 10 | 9.5 |
| 2.3 | 10 | 10 | 10 | 10 | 9 | 9 | 9.7 |
| 2.4 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 3.1 | 9 | 10 | 9 | 10 | 9 | 10 | 9.5 |
| 3.2 | 7 | 10 | 10 | 10 | 7 | 10 | 9 |
| 3.3 | 10 | 9 | 7 | 10 | 7 | 8 | 8.5 |
| 3.4 | 10 | 7 | 10 | 8 | 7 | 7 | 8.2 |
| 3.5 | 10 | 10 | 8 | 8 | 9 | 10 | 9.2 |
| 4.1 | 9 | 10 | 9 | 10 | 10 | 9 | 9.5 |
| 4.2 | 10 | 10 | 10 | 10 | 9 | 10 | 9.8 |
| 4.3 | 10 | 9 | 9 | 10 | 7 | 9 | 9 |
| 4.4 | 9 | 9 | 10 | 8 | 9 | 9 | 9 |

```
=ROUND(AVERAGE(B2:G2),1)
```

**Post Evaluation Summary**

From the evaluations, our prototype is very strong in conveying the information that we wanted it to. We only fell below an 8.0 average on one question, which would be our weakness. This category was about that navigability of the prototype without a mouse. This could be improved by fixing the tab orders for all the forms in the prototype as well as setting accept and escape buttons for quicker shortcuts. Our best scoring categories got a 10 on average, which is the maximum score possible. This was achieved in not using color to tell information, as well as, not putting important information into moving

objects. These scores were achieved because all the information listed on our prototype was clearly written, and never just color based. In addition, no information was ever stored in moving objects making readability great for our users.