



Eagles Housemate Manager

Team Members

219046379, Mpendulo Mthembu

Implementation & Deployment Document

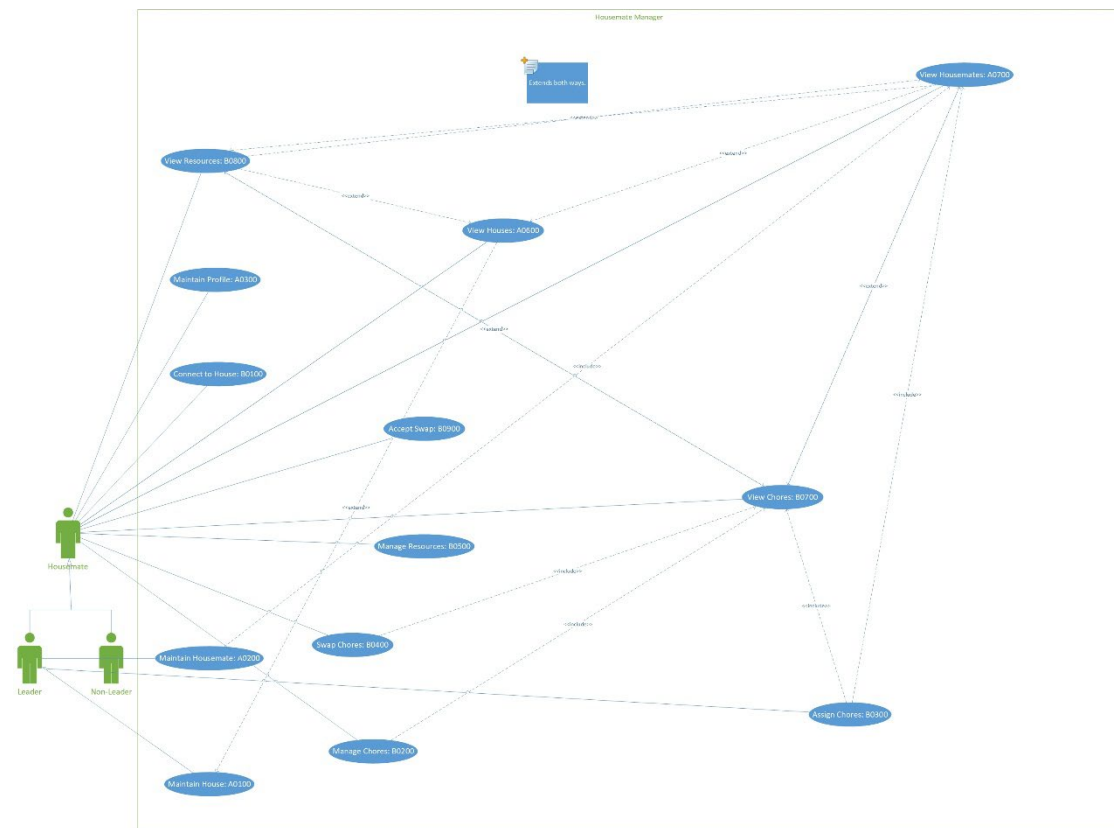
Document Date: 11/10/2022

TABLE OF CONTENTS

1	FUNCTIONAL REQUIREMENTS.....	3
1.1	Use Case Model.....	3
1.2	Use Case Glossary and Implementation Responsibilities	4
2	USE CASE NARRATIVES & UI DESIGNS.....	5
2.1	Team UI Guidelines	5
2.2	Navigation & Control	5
2.3	Team Design for.....	7
2.3.1	Login / Logout	7
2.3.2	Dashboard / Landing Page / Main Page	7
2.4	UI Designs & Use Case Narratives for the Implemented Use Cases.....	9
2.4.1	Designed by Mpendulo Mthembu	9
2.4.2	Designed by Mpendulo Mthembu	13
2.4.3	Designed by Mpendulo Mthembu	15
2.4.4	Designed by Mpendulo Mthembu	18
2.4.5	Designed by Mpendulo Mthembu	20
2.4.6	Designed by Mpendulo Mthembu	24
3	DATA REQUIREMENTS	29
3.1	Implementation Ready Class Diagram	29
3.2	Implemented database structure	30
3.3	Test Data.....	32
4	TECHNICAL REQUIREMENTS	34
4.1	System Environment.....	34
4.2	Start-up Instructions	34
4.3	Logins & Passwords	34

1 FUNCTIONAL REQUIREMENTS

1.1 Use Case Model



1.2 Use Case Glossary and Implementation Responsibilities

Team Member Responsible: Mpendulo Mthembu	
Use Case Id	Use Case Name
A0200	Maintain Housemates
A0300	Maintain Profile
B0200	Manage Chores
B0300	Assign Chores
Reports / Views / Queries	
A0700	View Housemates
B0800	View Resources

2 Use Case Narratives & UI Designs

2.1 Team UI Guidelines

The design of the system is minimalistic and does not use unnecessary UI objects or colors but includes those elements that are necessary for the user to perform their operations. The system has a dashboard which allows users to access all other functions of the system unless otherwise restricted. Functions such as to maintain profile or to assign chores are accessed by selecting the respective link on the dashboard which triggers a window to pop up which allows the user to perform that function.

Dashboard

The dashboard is the first page that the user sees after having registered/logged-in when opening the system and is where all links to other functions of the system are. The user can only select those links to which they have access to. For instance, only a housemate leader can add and/or remove other housemates therefore the Maintain Housemates page is restricted to leader housemates.

2.2 Navigation & Control

Navigation & control

The system uses tabs which separates, and groups related functions without the user having to leave a particular page. The system also uses list controls and check boxes for functions that require the use of information that already exist for easy retrieval and access.

Support for User Efficiency

The system's simplicity minimizes clustering thus ensuring efficient use of the system, use of UI objects that optimizes user's recognition rather than recall reduces users' time spent on any part of the system.

Recognition rather than recall

The system uses list controls and similar UI objects so that the user selects from required data which saves the user the time and effort of remembering detail required to perform database queries. Tooltips and default values are also used in appropriate UI objects.

Aesthetic Design

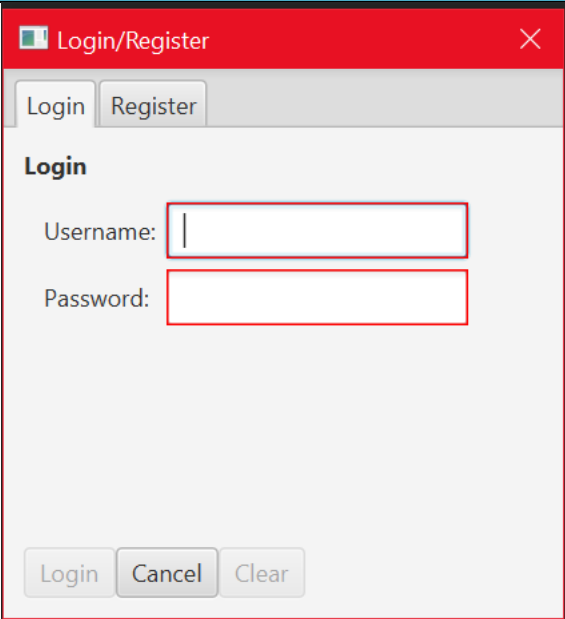
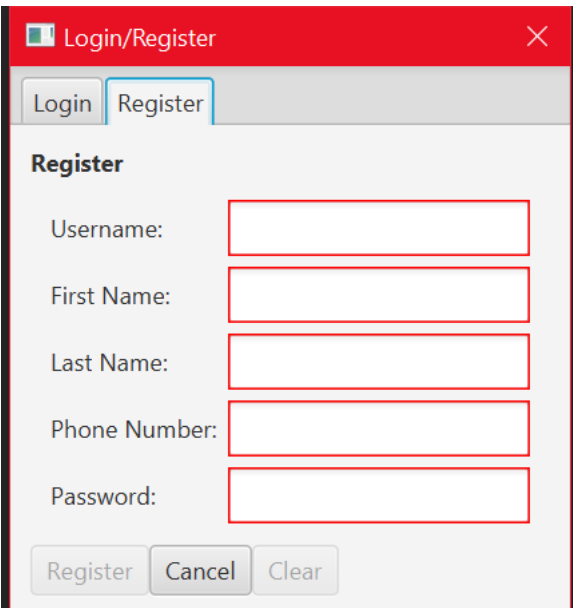
The user interface uses minimalistic UI objects and colors and makes effective use of spaces around UI objects which optimizes the user's use of the system.

Error prevention / recovery

Confirmation dialog boxes are used to make sure a user performed what they intend to do. The user of UI objects as list controls or similar so that the user selects required detail to reduce input-based mistakes. The system highlights risky operations to the user for instance by using a red border on the remove/delete button of housemates/chores etc.

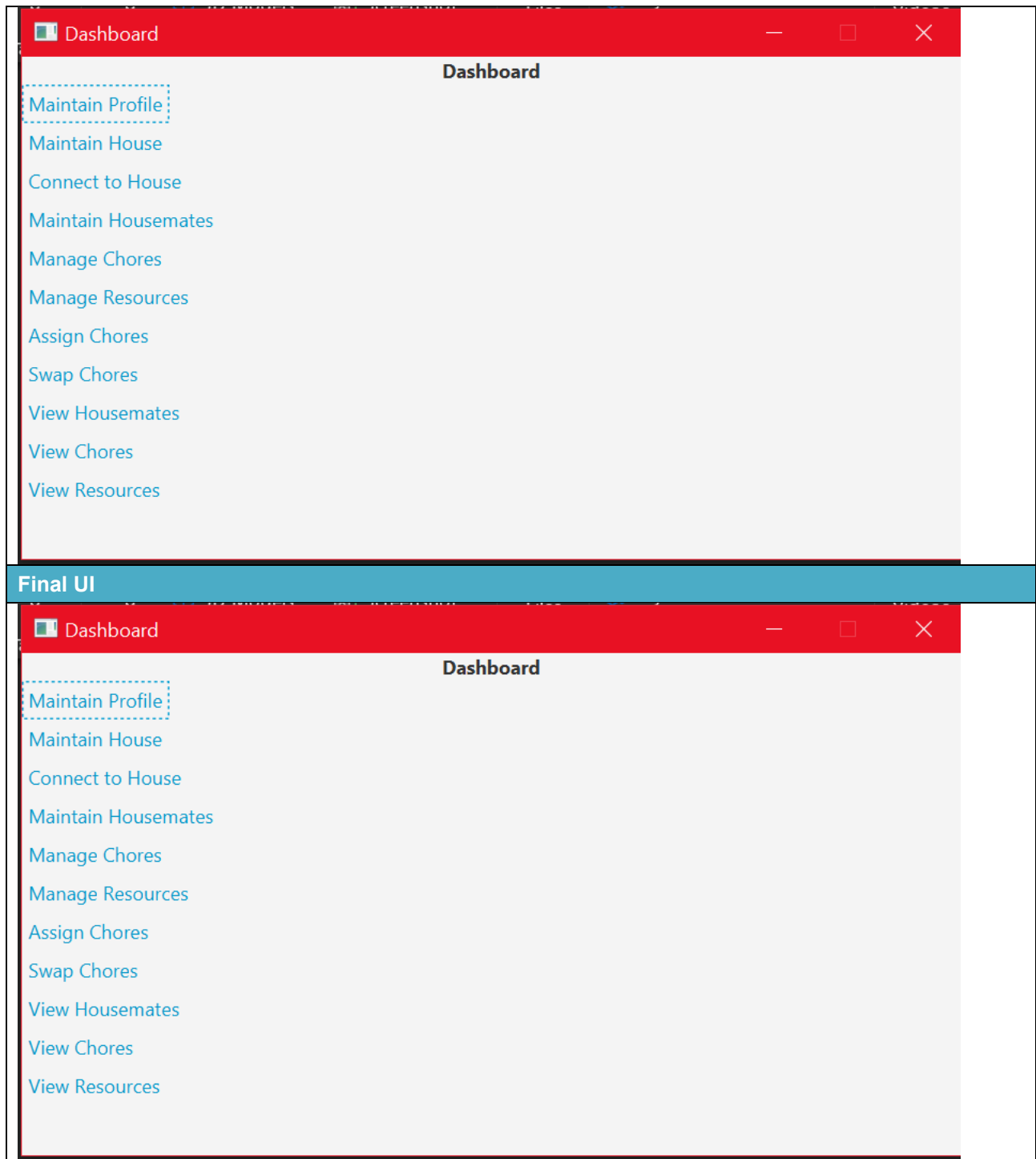
2.3 Team Design for

2.3.1 Login / Logout

Initial UI design	
Final UI	
 <p>The final UI design for the Login/Register window is shown. It features a red title bar with the text "Login/Register" and a close button. Below the title bar are two tabs: "Login" and "Register". The "Login" tab is selected, and the "Login" section is displayed. It contains two input fields: "Username:" and "Password:". At the bottom, there are three buttons: "Login", "Cancel", and "Clear".</p>	
 <p>The final UI design for the Login/Register window is shown. It features a red title bar with the text "Login/Register" and a close button. Below the title bar are two tabs: "Login" and "Register". The "Register" tab is selected, and the "Register" section is displayed. It contains five input fields: "Username:", "First Name:", "Last Name:", "Phone Number:", and "Password:". At the bottom, there are three buttons: "Register", "Cancel", and "Clear".</p>	

2.3.2 Dashboard / Landing Page / Main Page

Initial UI design



2.4 UI Designs & Use Case Narratives for the Implemented Use Cases

2.4.1 Designed by Mpendulo Mthembu

Use Case ID	Use Case Name
A0100	Maintain Housemates
Primary Business Actors	Other participating Actors
Housemate (Leader)	
Description	A house leader can add/edit/remove a housemate from a particular house.
Pre-Conditions	A connection must already have been made to a particular house. The housemate must be a house leader.
Triggers	A housemate selects to maintain housemates.
Post-Conditions	A housemate would then be added/edited/removed.
Basic Flow of Events	Maintain housemates: 1. A maintain housemates page is displayed Add: 1.1 The housemate selects the add tab 1.2 The add housemate tab is displayed 1.3 The housemate enters the housemate details. 1.4 The housemate selects to add the housemate 1.5 Appropriate system feedback is displayed.
Alternate Flow of events	Edit: 1.1 The housemate selects the edit tab 1.2 The edit housemate tab is displayed. 1.3 The housemate selects a housemate from the table. 1.4 The housemate selects to edit housemate. 1.5 Appropriate system feedback is displayed. Remove: 1.1 The housemate selects the remove tab 1.2 The remove housemate tab is displayed. 1.3 The housemate selects a housemate from the table. 1.4 The housemate selects to remove housemate. 1.5 A confirmation dialog is displayed.
Initial UI design	

Maintain Housemates

AddRemove

Details

Firstname: Jane

Lastname: Doe

Phone Number: 0813837673

Password: tempPassword

System feedback...

	ID	Firstname	Lastname
*			

Add

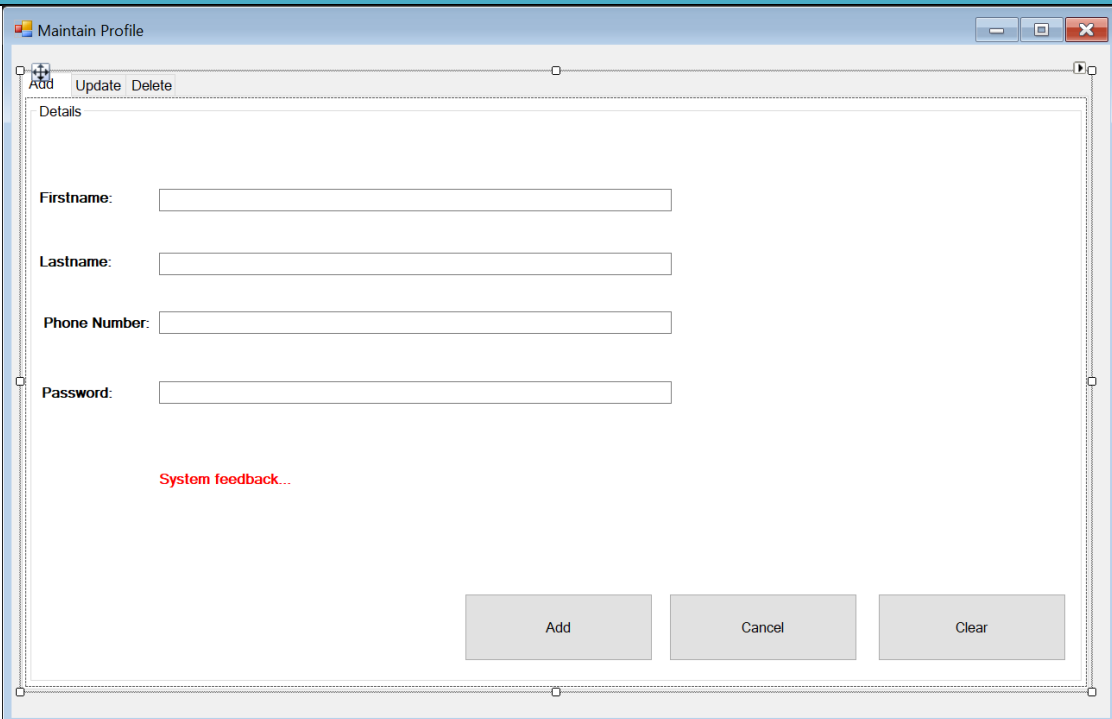
Cancel

Clear

Final UI

[illegible]

2.4.2 Designed by Mpendulo Mthembu

Use Case ID	Use Case Name	
A0300	Maintain Profile	
Primary Business Actors		Other participating Actors
Housemate (Non-Leader)		Housemate (Leader)
Description	A housemate can update/delete their own profile.	
Pre-Conditions	A profile must already exist	
Triggers	A housemate selects to maintain their profile.	
Post-Conditions	A housemate updates/deletes their profile.	
Basic Flow of Events	Maintain profile: 1. The maintain profile page is displayed.	
	Update: 1.1 The housemate selects the update tab. 1.2 The edit profile tab is displayed. 1.3 A housemate changes relevant detail. 1.4 A housemate then selects to update their profile. 1.5 Appropriate system feedback is displayed.	
Alternate Flow of events	Delete: 1.1 The housemate selects the delete tab. 1.2 The delete profile tab is displayed. 1.3 A housemate then selects to delete their profile. 1.4 A confirmation dialog is displayed. 1.5 Appropriate system feedback is displayed.	
Initial UI design		
		
Final UI		

Maintain Profile

EditDelete

Edit Profile

Housemate ID:1

Username:bill

First Name:Bill

Last Name:Bob

Phone Number:0894748632

Password:1234

Leader☒

EditCancelClear

Maintain Profile

EditDelete

Delete Profile

Housemate ID:1

Username:bill

First Name:Bill

Last Name:Bob

Phone Number:0894748632

Password:

Leader☒

RemoveCancelClear

2.4.3 Designed by Mpendulo Mthembu

Use Case ID	Use Case Name	
B0200	Manage Chores	
Primary Business Actors		Other participating Actors
Housemate (Non-Leader)		Housemate (Leader)
Description	A housemate can add/update/remove their chores, such as marking the chore as having been completed in case of editing.	
	A housemate leader can assign the chore to themselves (self-assign) as they create it or a “dangling” chore which can be later assigned.	
	A housemate leader can edit chores assigned to other housemates.	
Pre-Conditions	A connection must already have been made to a particular house.	
Triggers	A housemate selects to manage chores.	
Post-Conditions	A chore is then added/updated/removed.	
Basic Flow of Events	Manage chores: 1. A manage chores page is displayed.	
	Add: 1.1 The housemate selects the add tab. 1.2 The add chore tab is displayed. 1.3 A housemate enters the relevant detail. 1.4 A housemate then selects to add a chore. 1.5 Appropriate system feedback is displayed.	
Alternate Flow of events	Update: 1.1 The housemate selects the update tab. 1.2 The edit chore tab is displayed. 1.3 The housemate selects a chore from the table for which they would like to update. 1.4 A housemate changes relevant detail. 1.5 A housemate selects to update a chore. 1.6 Appropriate system feedback is displayed.	
	Remove: 1.1 The housemate selects the remove tab. 1.2 The remove chore tab is displayed. 1.3 The housemate selects the housemate from the table. 1.4 The housemate chooses to remove a chore. 1.5 A confirmation dialog is displayed. 1.6 Appropriate system feedback is displayed.	
Initial UI design		

Manage Chores

AddUpdateRemove

Details

Resources:

☐ checkedListBox1

House Area:

Description:

System feedback...

Add

Cancel

Clear

Final UI

Manage Chore

Add

Edit

Remove

Add Chore

Chore ID:

Resources:

Area Name:

Description:

Description not entered.

Completed:

Date Completed:

Chore ID	Area Name	Description	Completed	Date Completed
3	Kitchen	A chore for the ...	0	
5	Veranda	A chore for the ...	0	
7	Backyard	A chore for the ...	0	
9	Left roof	A chore for the ...	0	
14	Stairs	A chore for the ...	0	
15	Right roof	A chore for the ...	0	
16	Lawn	A chore for the ...	0	
17	Dinning room	A chore for the ...	0	
18	Drive way	A chore for the ...	0	
19	Main bed room	A chore for the ...	0	
20	Guest bed room	A chore for the ...	0	

Self assign

Add

Cancel

Clear

Manage Chore

Add

Edit

Remove

Edit Chore

Chore ID:

Resources:

Area Name:

Description:

Completed:

Date Completed:

Chore ID	Area Name	Description	Completed	Date Completed
3	Kitchen	A chore for the ...	0	
5	Veranda	A chore for the ...	0	
7	Backyard	A chore for the ...	0	
9	Left roof	A chore for the ...	0	
14	Stairs	A chore for the ...	0	
15	Right roof	A chore for the ...	0	
16	Lawn	A chore for the ...	0	
17	Dinning room	A chore for the ...	0	
18	Drive way	A chore for the ...	0	
19	Main bed room	A chore for the ...	0	
20	Guest bed room	A chore for the ...	0	

Edit

Cancel

Clear

Page 17 of 34

Manage Chore

Add

Edit

Remove

Remove Chore

Chore ID:

Resources:

Area Name:

Description:

Completed:

☐

Date Completed:

Chore ID	Area Name	Description	Completed	Date Completed
3	Kitchen	A chore for the ...	0	
5	Veranda	A chore for the ...	0	
7	Backyard	A chore for the ...	0	
9	Left roof	A chore for the ...	0	
14	Stairs	A chore for the ...	0	
15	Right roof	A chore for the ...	0	
16	Lawn	A chore for the ...	0	
17	Dinning room	A chore for the ...	0	
18	Drive way	A chore for the ...	0	
19	Main bed room	A chore for the ...	0	
20	Guest bed room	A chore for the ...	0	

Remove

Cancel

Clear

2.4.4 Designed by Mpendulo Mthembu

Use Case ID	Use Case Name	
B0300	Assign Chores	
Primary Business Actors		Other participating Actors
Housemate (Leader)		
Description	A house leader can assign chores to a particular housemate.	
Pre-Conditions	A connection must already have been made to a particular house. The housemate must be a house leader. A chore must already exist and unassigned to another housemate.	
Triggers	A housemate selects to assign house chores.	
Post-Conditions	Chore is then assigned to a particular housemate.	
Basic Flow of Events	<div>1. An assign chores page is displayed.</div> <div>2. The housemate selects chore(s) and a housemate from the lists to assign to a housemate.</div> <div>3. The housemate selects to assign chore(s).</div> <div>4. Appropriate system feedback is displayed.</div>	
Alternate Flow of events		
Initial UI design		

Assign Chores

Details

Firstname:

Lastname:

Description:

System feedback...

	HousemateID	Firstname	Lastname	Phone Number
*				

	ChoreID	HouseArea	Description	HousemateID	IsCompleted
*					

Assign

Cancel

Final UI

Assign Chore

Assign Chores

Housemate ID:

Username:

First Name:

Last Name:

Phone Number:

Password:

Leader

☐

Chore ID:

Resources:

Area Name:

Description:

Completed:

☐

Date Completed:

Assign

Cancel

Housemates

Housemate ID	Username	First Name	Last Name	Phone Number
1	bill	Bill	Bob	0894748632
2	junior	Junior	Bob	0838263872
4	benn	Benn	Stealer	0897987897
5	zach	Zachary	Russell	0897987498
6	harry	Harry	Garner	0773823848
7	eve	Eve	Black	0727473822
10	joe	Joe	Parkinson	0639493234
11

Chores

Chore ID	Area Name	Description	Completed	Date Completed
No content in table				

2.4.5 Designed by Mpendulo Mthembu

Use Case ID	Use Case Name
A0700	View Housemates
Primary Business Actors	Other participating Actors
Housemate (Leader)	Housemate (Non-Leader)
Description	A housemate can view housemates and has the option to view related detail such as chores and resources bought using B0700 and B0800 respectively.
Pre-Conditions	A connection must already have been made to a particular house.
Triggers	A housemate selects to view housemates.
Post-Conditions	A table of housemates and the option to view related details is displayed.
Basic Flow of Events	1. A view housemates page is displayed.
Alternate Flow of events	<p>View resources:</p> <p>1.1 Housemates selects a housemate from table</p> <p>View housemate chores:</p> <p>1.1.1 Housemate selects to view that housemate's chore</p> <p>1.1.2 A view housemate chores page is displayed.</p> <p>View housemate resources:</p> <p>1.2.1 Housemate selects to view that housemate's resources</p> <p>1.2.2 A view housemate resources page is displayed.</p>

Initial UI design

The UI design shows a window titled "View Housemates". It contains a table with the following columns: HousemateID, Firstname, Lastname, and Phone Number. The first row of the table is highlighted in grey and contains an asterisk (*). Below the table, there are two buttons: "View Resources" and "View Chores".

View Housemates

View Housemates

Housemate ID	Username	First Name	Last Name	Phone Number	Password	Leader
1	bill	Bill	Bob	0894748632	1234	1
2	junior	Junior	Bob	0838263872	2345	0
4	benn	Benn	Stealer	0897987897	3456	0

View Resources

View Chores

Cancel

[illegible]

Cancel

View resources for Bill Bob

View resources for Bill Bob

Resource Name	Is Finished	Housemate ID
mop	0	1
sunlight	0	1

View Housemates

View Chores

Cancel

2.4.6 Designed by Mpendulo Mthembu

Use Case ID	Use Case Name
B0800	View Resources
Primary Business Actors	Other participating Actors
Housemate (Leader)	Housemate (Non-Leader)
Description	A housemate can view resource and has the option to view related details such as chores associated with that resource and housemate who bought that resource using B0700 and A0700 respectively.
Pre-Conditions	A connection must already have been made to a particular house.
Triggers	A housemate selects to view resources.
Post-Conditions	A table of resources and the option to view related detail is displayed.
Basic Flow of Events	1. A view resources page is displayed.
Alternate Flow of events	<p>View resources:</p> <p>1.1 Housemates selects a resource from table</p> <p>View resource's chores:</p> <p>1.1.1 Housemate selects to view that chores that use that resource</p>

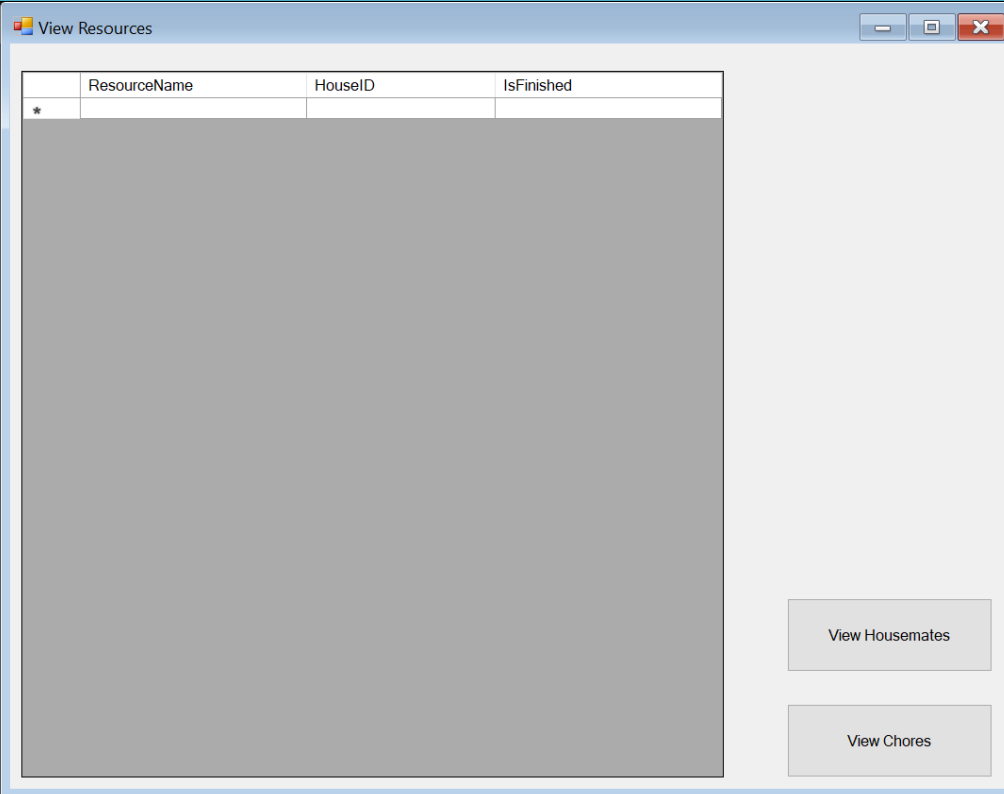
1.1.2 A view resource chores page is displayed.

View resource's housemate:

1.2.1 Housemate selects to view housemate who bought that resource.

1.2.2 A view resource housemate page is displayed.

Initial UI design



Final UI

View Resources

View Resources

Resource Name	Is Finished	Housemate ID
brush	0	2
mop	0	1
Sunlight	0	1

View Housemates

View Chores

Cancel

x

[View Resources](#)[illegible]

View Housemates View Chores Cancel

[illegible]

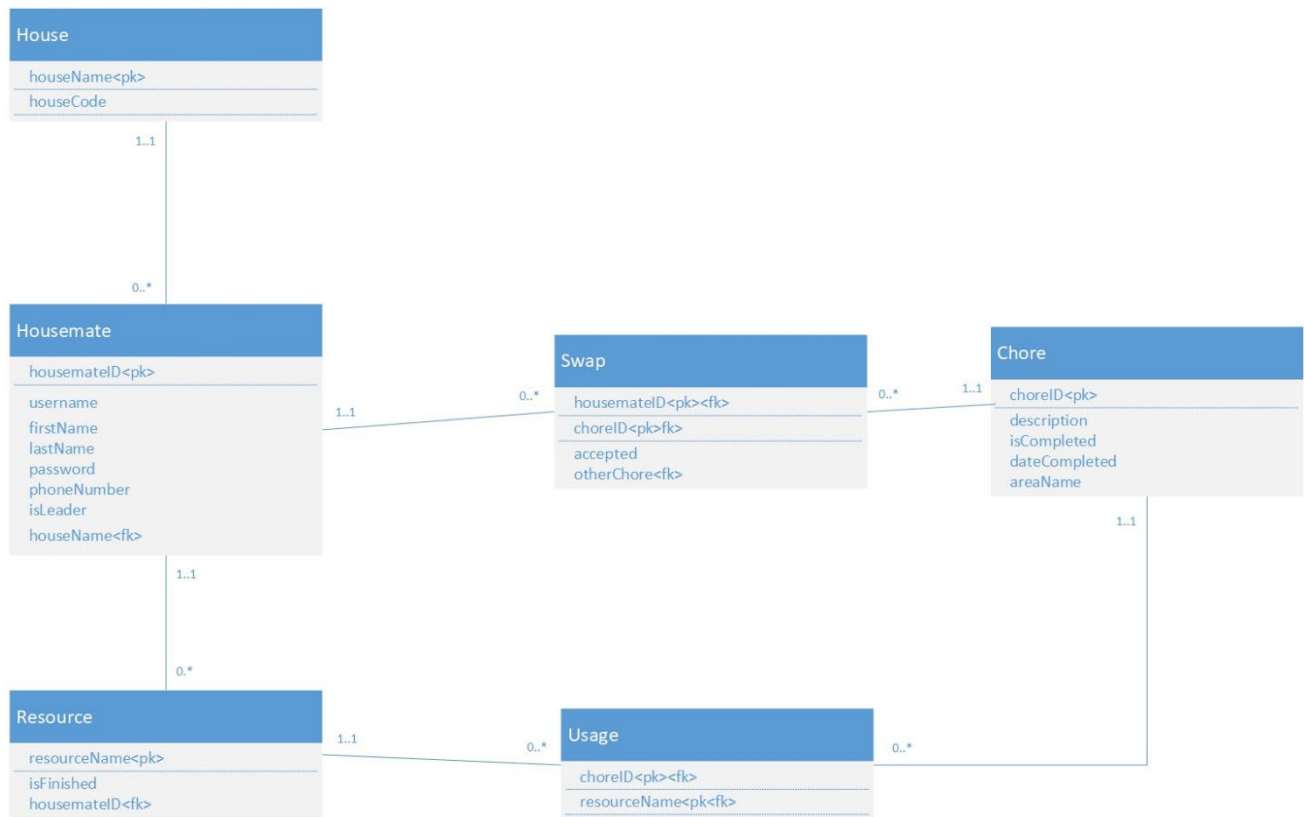
Cancel

[illegible]

Cancel

3 DATA REQUIREMENTS

3.1 Implementation Ready Class Diagram








3.2 Implemented database structure

We have used **MS SQL** to create our database in. The name of the database for our project is **HMS4**



dbo.Chore

Columns

-  choreID (PK, int, not null)
-  description (varchar(255), not null)
-  isCompleted (bit, not null)
-  dateCompleted (date, null)
-  areaName (varchar(30), not null)









dbo.House

Columns

-  houseName (PK, varchar(30), not null)
-  houseCode (char(4), not null)




dbo.Housemate



Columns


-  housemateID (PK, int, not null)
-  username (varchar(30), not null)
-  firstName (varchar(30), not null)
-  lastName (varchar(30), not null)
-  phoneNumber (char(10), not null)
-  isLeader (bit, not null)
-  password (varchar(30), not null)
-  houseName (FK, varchar(30), null)

dbo.Swap


Columns


-  housemateID (PK, FK, int, not null)
-  choreID (PK, FK, int, not null)
-  otherChore (FK, int, null)

  **dbo.Usage**

 **Columns**

 choreID (PK, FK, int, not null)

 resourceName (PK, FK, varchar(30), not null)

 housemateID (PK, FK, int, not null)

3.3 Test Data

Chore										
	1	2	3	4	5	6	7	8	9	10
choreID	3	5	7	9	14	15	16	17	18	19
description	A chore for the kitchen .	A chore for the veranda .	A chore for the backyard .	A chore for the left side of the roof.	A chore for the stairs .	A chore for the right side of the roof.	A chore for the lawn.	A chore for the dinning room.	A chore for the drive way.	A chore for the main bed room .
isCompleted	0	0	0	0	0	0	0	0	0	0
dateCompleted	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
areaName	Kitchen	Veranda	Backyard	Left roof	Stairs	Right roof	Lawn	Dinning room	Drive way	Main bed room

Housemate										
	1	2	3	4	5	6	7	8	9	10
housemateID	1	2	4	5	6	7	10	11	12	13
username	bill	junior	benn	zach	harry	eve	joe	ava	mad	max
firstName	Bill	Junior	Benn	Zach	Harry	Eve	Joe	Ava	Mad	Max
lastName	Bob	Bob	Stealer	Russel	Garner	Black	Parkinson	Harding	Reynolds	Ryan
phoneNumber	0894748632	0838263872	0897987897	0897987498	0773823848	0727473822	0639493234	0622838483	0782837473	0873828732
isLeader	1	0	0	0	0	0	0	0	0	0
password	1234	2345	3456	789734	dhjds	eojuwe8	iowe798	987ewtyug	iyiigy*&	98432
houseName	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Resource										
	1	2	3	4	5	6	7	8	9	10
resourceName	Bags	Bags	Bags	Bin	Bin	Broom	Broom	Broom	Broom	Broom
isFinished	0	0	0	0	0	0	0	0	0	0
housemateID	4	6	10	7	13	1	4	10	11	12

Swap										
	1	2	3	4	5	6	7	8	9	10
housemateID	1	2	2	4	5	6	7	10	11	12
choreID	3	7	14	9	18	5	16	15	17	19
otherChore	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Usage										
	1	2	3	4	5	6	7	8	9	10

choreID	3	9	9	14	15	15	16	17	19	19
resourceName	Broom	Bags	Broom	Brush	Bags	Broom	Brush	Bags	Broom	Bin
housemateID	1	4	4	2	10	10	7	11	12	12

4 TECHNICAL REQUIREMENTS

4.1 System Environment

The system runs on Windows 10. The WRR.bat file used to run the system uses the JAVA_HOME environment variable therefore the operating system should have java installed and the JAVA_HOME environment variable set. e.g., "C:\Program Files\Java\jdk-17.0.1" excluding the bin folder.

4.2 Start-up Instructions

The system can be run by double clicking the **WRR.bat** file which is located at **WRR\out\artifacts\WRR.jar** within the project. The system can also be run from IntelliJ by importing the solution file.

4.3 Logins & Passwords

	housemateID	username	firstName	lastName	phoneNumber	isLeader	password	houseName
1	1 ADMIN	bill	Bill	Bob	0894748632	1	1234	NULL
2	2	junior	Junior	Bob	0838263872	0	2345	NULL
3	4	benn	Benn	Stealer	0897987897	0	3456	NULL
4	5	zach	Zachary	Russell	0897987498	0	789734	NULL
5	6	harry	Harry	Garner	0773823848	0	dhjds	NULL
6	7	eve	Eve	Black	0727473822	0	eoIUwe8	NULL
7	10	joe	Joe	Parkinson	0639493234	0	iowe798	NULL
8	11	ava	Ava	Harding	0622838483	0	987ewtyug	NULL
9	12	mad	Maddison	Reynolds	0782837473	0	iyiigy*&	NULL
10	13	max	Max	Ryan	0873828732	0	98432	NULL