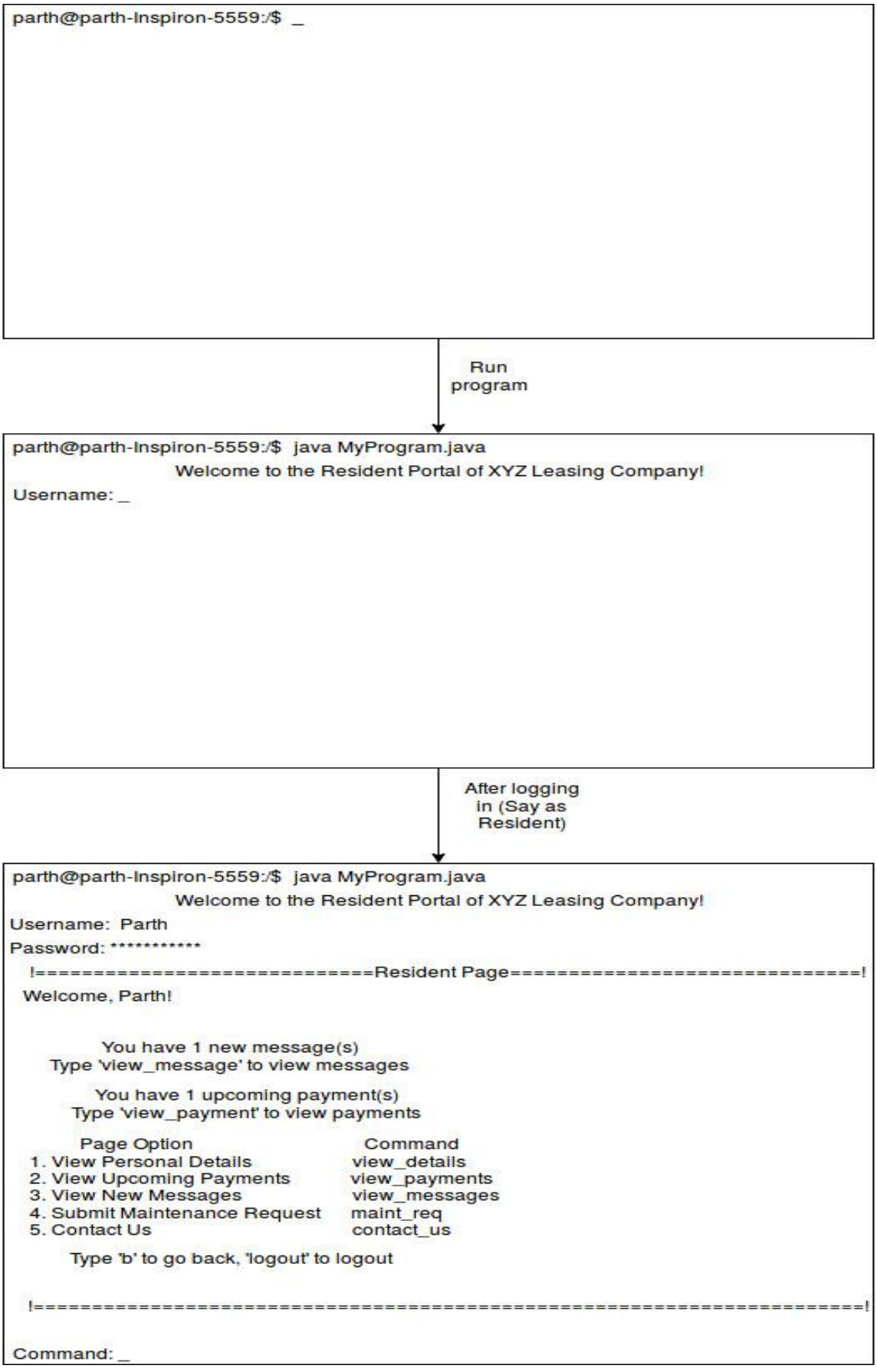
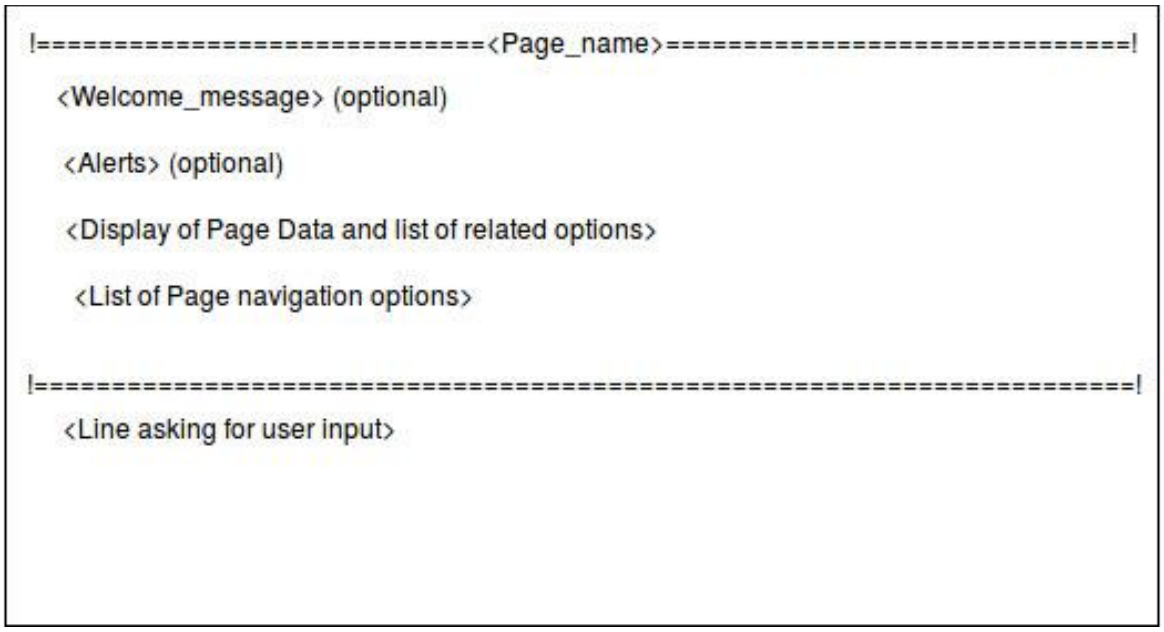
1. **Name:** Parth Thakkar
2. **Title:** Resident Portal
3. **Project Summary:** A housing complex website which helps residents pay the rent and submit maintenance requests. It would also keep track of the same. We will set up for 1 resident in (say) a studio apartment.
4. **Project Requirements:**

Req. No.	Client Req. No.	Requirements
UR-7	7.	As a Resident, I can submit maintenance requests
UR-8	8.	As a LM, I can access the maintenance requests
UR-9	7.	As a Resident, I can track the status of my maintenance requests
UR-10	8.	As a LM, I can update the status of the maintenance requests
UR-11	7.	As a Resident, I can get alerts for status updates on maintenance requests

5. **UI Mockups:** I'm using a simple command line UI. The following is an example of what will happen if we login as a resident.


parth@parth-Inspiron-5559:/$ _
Run program
parth@parth-Inspiron-5559:/$ java MyProgram.java
Welcome to the Resident Portal of XYZ Leasing Company!
Username: _
After logging in (Say as Resident)
parth@parth-Inspiron-5559:/$ java MyProgram.java
Welcome to the Resident Portal of XYZ Leasing Company!
Username: Parth
Password: ***********
!==========Resident Page==========!
Welcome, Parth!
You have 1 new message(s)
Type 'view_message' to view messages
You have 1 upcoming payment(s)
Type 'view_payment' to view payments
Page Option
Command
1. View Personal Details
view_details
2. View Upcoming Payments
view_payments
3. View New Messages
view_messages
4. Submit Maintenance Request
maint_req
5. Contact Us
contact_us
Type 'b' to go back, 'logout' to logout
!==========!
Command: _

The general layout of a 'page' would be as follows:



If we're using a MVC Architecture, this can be used as a template for the Template Design Pattern, which could be used in classes designated for creating and displaying the View.

The template could have the following order of operations:

a. Display the page name
b. Display a welcome message
c. Display Alerts. These could be new messages, upcoming payment dates, updates on maintenance request status, etc. **Give options along with alerts.**
d. Display Page Data and a list of related options. For example, if the page is of a Maintenance Request Form, this could contain the names of the fields (With specification of whether the field is required or not), values of entered fields.
e. Display List of page navigation options: These could be options for going back (with redirection errors if required), logging out (with errors if required, for example if the user is in the middle of filling out a form there'd be an error), submitting the form, etc.
f. Line asking for user input

Example Use case: Say, the Resident in question wants to edit their Personal Details.

- Then, they can give the command 'view_details' and view their details. These details will be well organized so that it doesn't create a mess on the screen (to the best of my abilities).
- A 'page' will be displayed. The 'Page Data and Options' part of the page will contain the Resident's details with the option of choosing which detail they want to edit by typing in the field name.
- If they choose to edit a detail, their input will be taken, stored in the system and then they'll be redirected to the page which holds the details, possibly edited.

This can happen provided the Resident has permission to edit the specific detail. If not, then an error message will be displayed and the Resident will be redirected to the page with their details.

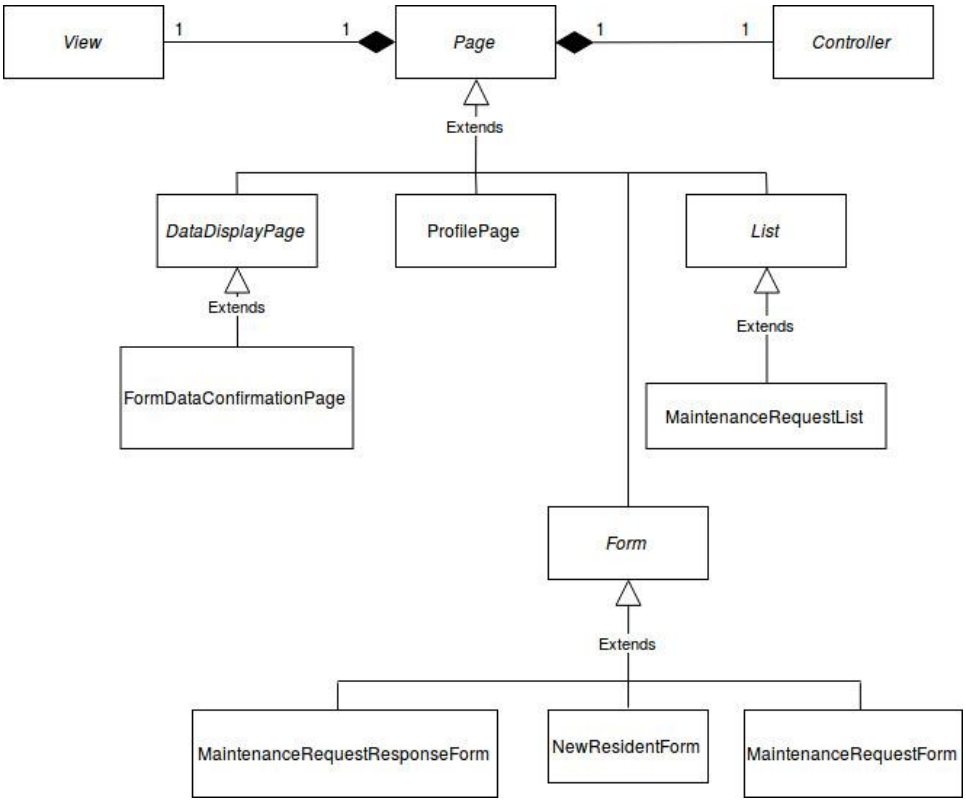
Actors: Resident, Leasing Manager (LM)

Forms we'll be dealing with here:

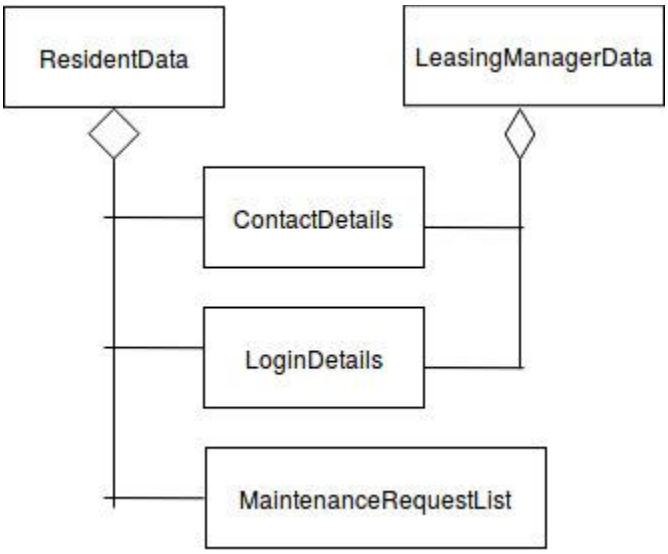
a. New Resident Form: This form will be filled out by the LM in order to have the Resident be a part of the system. The LM can give the Resident a Username and Password. The code will have the Username and Password of the LM inbuilt. The LM will have options to cancel or submit.
b. Maintenance Request Form: This form will be used by the Resident in order to communicate a maintenance issue to the LM. Since this form is not saved until submitted, it is assumed that the Resident wants to fill out all the fields before leaving the page and so the command prompt will ask the Resident to fill out each field one after the other. The Resident will have the option to cancel or submit.
c. Maintenance Request Status Update Form: The LM can fill out this form as a response to the Maintenance Request Form.

Note: Instead of a drop down menu, here options will be given with instructions on how to choose one. Error messages will be shown if anything else is done.

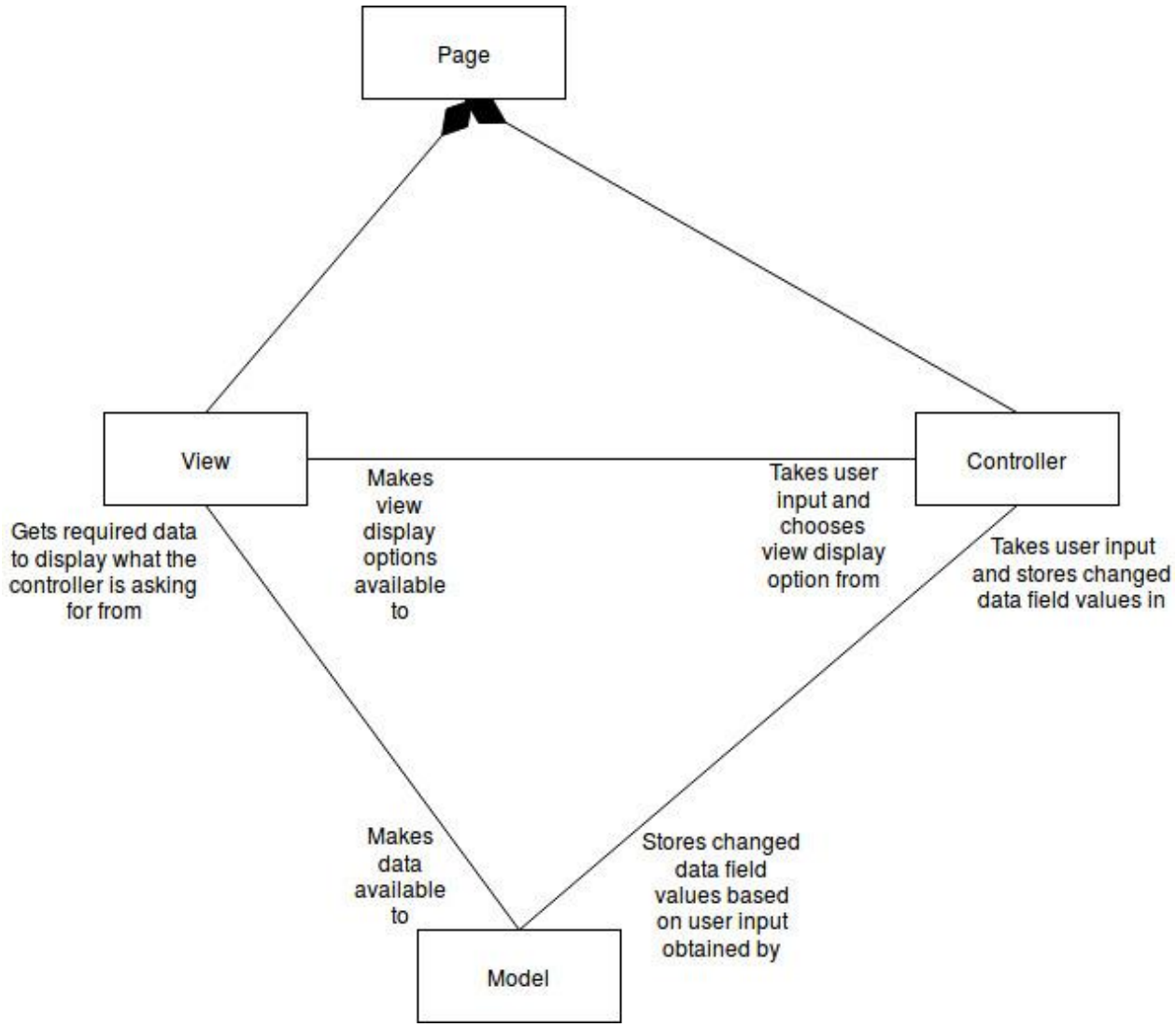
6. **Class Diagrams:** I have made the following class diagrams:



Class Diagram for Pages: The *View* class will use a Template Design Pattern for printing the correct view on the screen. The *Controller* class will contain the Options for each of the Pages and instructions for what to do if a specific option is selected.



Class Diagram for Model: The Organization of data in the Model. The Model isn't necessarily a class, so this whole organization of classes to store data is termed as the Model in the next diagram.



Holistic Class Diagram for MVC: The way the MVC would be organized. The Controller is in charge of taking input from the user and choosing what to display accordingly and the View is in charge of displaying what would come on the screen when the user gives their input according to the Controller.