Software Requirements Specification

for

Tenant Portal with Java

Prepared by

**Israt Jahan Mim 171-1392-042**

**Sourav Debnath 171-1543-042**

**Md. Tahmidur Rahman 172-1370-042**

**Shah Alvi Hossain 172-1427-042**

Submitted to AKM Bahalul Haque

15 Feb 2020

**Table of Contents**

[1. Introduction 1](#_Toc32729442)

[1.1 Purpose 1](#_Toc32729443)

[1.2 Document Conventions 1](#_Toc32729444)

[1.3 Intended Audience and Reading Suggestions 1](#_Toc32729445)

[1.4 Product Scope 1](#_Toc32729446)

[2. Overall Description 1](#_Toc32729447)

[2.1 Product Perspective 1](#_Toc32729448)

[2.2 Product Functions 1](#_Toc32729449)

[2.3 User Classes and Characteristics 2](#_Toc32729450)

[2.4 Operating Environment 2](#_Toc32729451)

[2.5 Design and Implementation Constraints 2](#_Toc32729452)

[2.6 User Documentation 2](#_Toc32729453)

[2.7 Assumptions and Dependencies 2](#_Toc32729454)

[3. External Interface Requirements 3](#_Toc32729455)

[3.1 User Interfaces 3](#_Toc32729456)

[3.2 Hardware Interfaces 3](#_Toc32729457)

[3.3 Software Interfaces 3](#_Toc32729458)

[3.4 Communications Interfaces 3](#_Toc32729459)

[3.5 Technology in use 3](#_Toc32729460)

[4. System Features 3](#_Toc32729461)

[4.1 Posting property vacancy 4](#_Toc32729462)

[4.2 Check individual profile 4](#_Toc32729463)

[4.3 General feed 4](#_Toc32729464)

[4.4 Tenant check posts 5](#_Toc32729465)

[4.5 Search for vacancy accordingly 5](#_Toc32729466)

[4.6 Payment process claim 5](#_Toc32729467)

[4.7 Verify due claim 6](#_Toc32729468)

[4.8 Check user account 6](#_Toc32729469)

[4.9 The super record 6](#_Toc32729470)

[4.10 Administer the feed 6](#_Toc32729471)

[5. Other Nonfunctional Requirements 7](#_Toc32729472)

[5.1 Performance Requirements 7](#_Toc32729473)

[5.2 Safety Requirements 7](#_Toc32729474)

[5.3 Security Requirements 7](#_Toc32729475)

[5.4 Software Quality Attributes 8](#_Toc32729476)

[6. Appendix 8](#_Toc32729477)

[7. Mockup 8](#_Toc32729478)

[8. Repository 8](#_Toc32729479)

# Introduction

## Purpose

The software is being built for CSE 299: Junior project design. It is a portal-based home rental system built completely out of Java. The purpose is to design and build something challenging in this little time window of 3 months.

## Document Conventions

To write this SRS IEEE standards are being followed, which is copyrighted by Karl E. Weigers.

## Intended Audience and Reading Suggestions

This document is intended for our course instructor AKM Bahalul Haque and the rest of the team members to help them guide through this project’s development processes. This SRS contains the product scope, overall description of the product which gives a better understanding what we want to do and how the things will work out, the various requirement’s, features, some nonfunctional requirements, what kind of system will be used to build it etc. The readers are advised to read it serially so that it is easier to understand and not miss anything

## Product Scope

For the CSE299 project the project idea had to be novel or really challenging to work with, which is why the idea to build a full java-based system is quite challenging for the team in general. The whole system will need to raw coded and painstakingly engineered. And managing the time is also a big challenge as the time limit is only 3 months.

# Overall Description

## Product Perspective

The main idea was to build an app and web app with the novel technology of flutter and show case the new ideas and scope that the technology can demonstrate. But the idea was quickly scraped and with nothing coming to mind the idea to develop something similar but portal based was decided and to add the complexity the whole thing has to be in java. From UI to backend all done in java is quite a task. But if it is managed perfectly with moderate working speed it is achievable.

## Product Functions

The system is sort of a portal that will allow the users to login and be an owner or a tenant, if an owner then can give their property for rent and if a tenant then can rent the property. The system will work as a middle man but system will also keep records form both sides. So that both parties can check on each other while renting or giving out rent. It ensures safety and reliability of each other. And the whole system is seen over by administrators. The admins can be government run organization etc.

## User Classes and Characteristics

The user class and character are mentioned below-

* User – originally in 2 parts owner and tenant. They both are normal users who will login to see the property market and maintain records.
* Admin – the admin is here only to look over the users. And if any problems come up between users, their job is to verify the claims and take necessary judgments upon them and fix the problems as they appear.

## Operating Environment

The products have one main environment to run on. It will run on computer supported by java. As the complete system is build with java. The system must have minimum capability to run java. The required library will be included and other data will be downloaded over the internet. And the database is accessed over the internet to maintain all time live and updated.

## Design and Implementation Constraints

The main implementation constraints are thought to be implementation of the google map on the system to show the property up for rent in the searched area. Other constraint is to implement all the quarries and they work in right manners.

## User Documentation

The system will be user friendly through straight forward UI design with minimum complexity and it eliminate the main concern of using a user manual to operate the portal. But upon completion adequate and easily understandable tutorials can be released. With all that there will be report a problem option which will allow the maintenance team to know what is wrong in the system or where the users are facing problems. But as the project is for education based the instructor will be guided to the demo by the developers for easy understandings.

## Assumptions and Dependencies

We are assuming that we will be able to use google map to pin out listings, google OAuth and open source API’s to make sure everything works. But if any how these are not available or any of these supports are held back, we will have problems with working on the development. Also, while working if any big changes are made in the environment that we are working on can be problematic. Any major future updates can effect on any of our development process.

# External Interface Requirements

## User Interfaces

Based on the users there will be multiple stages in the system which will fetch data over the internet. The UI is easy to understand as users will get options to work with according to their role in the system. If one is an admin, they will never need to post for rent of ask for bill clarification. Same as the users will never be able to have the admin powers to make drastic changes on another user’s records. The UI will be clean and easy to get by with easy on the eyes color selection. As the system will be a part of people’s lives the UI will be very interactive and dynamic to response.

## Hardware Interfaces

The system is purely java based so the computer must support java, and to fetch for real time data and updates to listings for rent internet is required. Mostly all the libraries can be in the system but the database will not be so internet is crucial.

## Software Interfaces

The system will be mostly a self-sustained system only requiring the java (JDK and JRE files in the running environment). But for database the system will user internet to connect with offsite database.

## Communications Interfaces

The system will use internet, as the database is currently set up as an FTP it will use FTP based connection and also HTTPS for the multiple API’s in use. The system might need all time internet to make sure the system is all time updated.

## Technology in use

According to instructors commands the technology in use are –

* Java
* JavaFx
* MySQL
* JFoenix
* MySql Connector
* Google 2.0 OAuth API
* Google Maps API
* PHP Mailer
* Java-json

# System Features

The system tenant portal has some basic and some key features to attract users.

## Posting property vacancy

4.1.1. Description and Priority

The user(owner) must be able to post which property is up for rent and their details along with it.

4.1.2. Functional Requirements

1. When the tenant is confirmed the owner must be able to remove the notice
2. Edit the post
3. Add pictures to the post

## Check individual profile

4.2.1. Description and Priority

Owner can see tenant and vice versa tenant can see owner profile as well.

Mostly see for any past records of harmful conduct or unpaid dues

4.2.2. Functional Requirements

* + 1. Check payment claim
    2. Calculate payment
    3. Updating payment
    4. Checking past dues
    5. See past records
    6. See profile

## General feed

4.3.1. Description and Priority

This is related to news feeds where the news is stacked one after another, the posts from owner will pile up on the feed

4.3.2. Functional Requirements

1. The feed is public everyone can see it
2. For tenants: option to rent which leads to other information
3. If owner deletes post it will be removed from feed also
4. Feed doesn’t show added picture

## Tenant check posts

4.4.1. Description and Priority

The tenants looking at the feed can go for the rent button where they will be guided to the owner’s profile and contact info with other information of the property

* + 1. Functional Requirements

1. See owner
2. Find information
3. Find map information

## Search for vacancy accordingly

4.5.1. Description and Priority

The tenants can also search by area or city if they don’t want to look at the live feed.

4.5.2. Functional Requirements

1. The search can be done by area, city etc.
2. The search result can be seen on map with all the vacant spots marked, this will be achieved with the help of google map API

## Payment process claim

4.6.1. Description and Priority

The system will keep a record of the bills paid out to the owners by the tenants. The tenant will submit the bill pay claim and the owner will accept it if he/she has received the money. This will allow to keep a permanent record for both sides

4.6.2. Functional Requirements

1. Along with transaction ID the tenant will send a request to be accepted by the owner that he/she has paid the bills
2. Usually the transaction ID are unique every time, so there is a very little window that 2 very random tenant will provide same transaction ID to be verified
3. As the owner confirm that the ID is correct the tenant profile will be updated saying all dues are clear
4. If the dues are not clear then it will show as not clear to the owners as well as any other owner who checks up on the tenant later

## Verify due claim

4.7.1. Description and Priority

In case the bill was paid and the owner didn’t accept the tenant will be able to appeal to the admin to take actions. Where the admin will be in direct contact with both sides and verify the claim and take actions

* + 1. Functional Requirements

1. The admin will see the reports made by the tenants
2. Make changes to the records if needed
3. Take actions against the owner or tenants

## Check user account

4.8.1. Description and Priority

The admin must be able to go through all the user accounts. They hold the power in the portal right after the developers to make changes.

4.8.2. Functional Requirements

1. The admin can suspend accounts
2. Admin can lift suspension and activate profile again
3. Admin can assign account role as well

## The super record

4.9.1. Description and Priority

The system will keep a type of permanent record for users. If they have any unpaid dues or tendency of paying bills late the system has the records for that and they can be easily accessed from profiles.

4.9.2. Functional Requirements

1. The records can not be changed by the account owner itself
2. If any mistakes were made the admin should be contacted and they will verify the claims and make changes if necessary
3. The edit or delete option can only be accessed by the admin for records

## Administer the feed

4.10.1. Description and Priority

The admins other duty is to admin the feed that is public to all. If anyone post something inappropriate or anything other then to-let or add picture to the post that is not relevant then they can take actions

4.10.2. Functional Requirements

1. Take actions on irrelevant posts
2. Delete posts
3. If post is deleted from feed it should also be removed from owner profile

# Other Nonfunctional Requirements

## Performance Requirements

* The system must have independent log in system
* Password reset feature
* Different options for admin and user login
* User should be able to contact admin easily
* The system should be aware of who is who- like the owner should not be able to rent his/her on property.
* The admin power should be only be accessed by admins.
* Every move made by the admins and users should be in logged to see who made the changes.
* Suspension/ban should be removeable
* The user and admin must have sessions to maintain security
* It must not save excess cookies for that reasons it can show wrong data sometimes
* Make sure right data is being fetched with each query
* Make sure multiple accounts can’t own same property to rent out

## Safety Requirements

Posting the vacancy online is a kind of risk to owner itself when the place is vacant for long time it can be harmful in many ways. And also, the people who are visiting the area or the building needs to be careful so that they don’t go alone or fall into a tap set up by fishing parties. The system will have to follow the policies set up by the government for domestic rental purposes. The system is maintained by an admin so contacting the admin if any problems occurs.

## Security Requirements

For main login feature API will be used to verify the login via the emails. And every account must be registered with their NID, and each NID holder will have only one account. This will deny any foul play. Use of NID ensures the identity of the users and rest assured the sensitive information are all be kept under secure data base.

## Software Quality Attributes

* The portal is for keeping a clear record between users and admins are to insure the validity of the records
* Insure backed up records
* Secure database

# Appendix

API: Application programming interface

IEEE: Institute of Electrical and Electronics Engineers

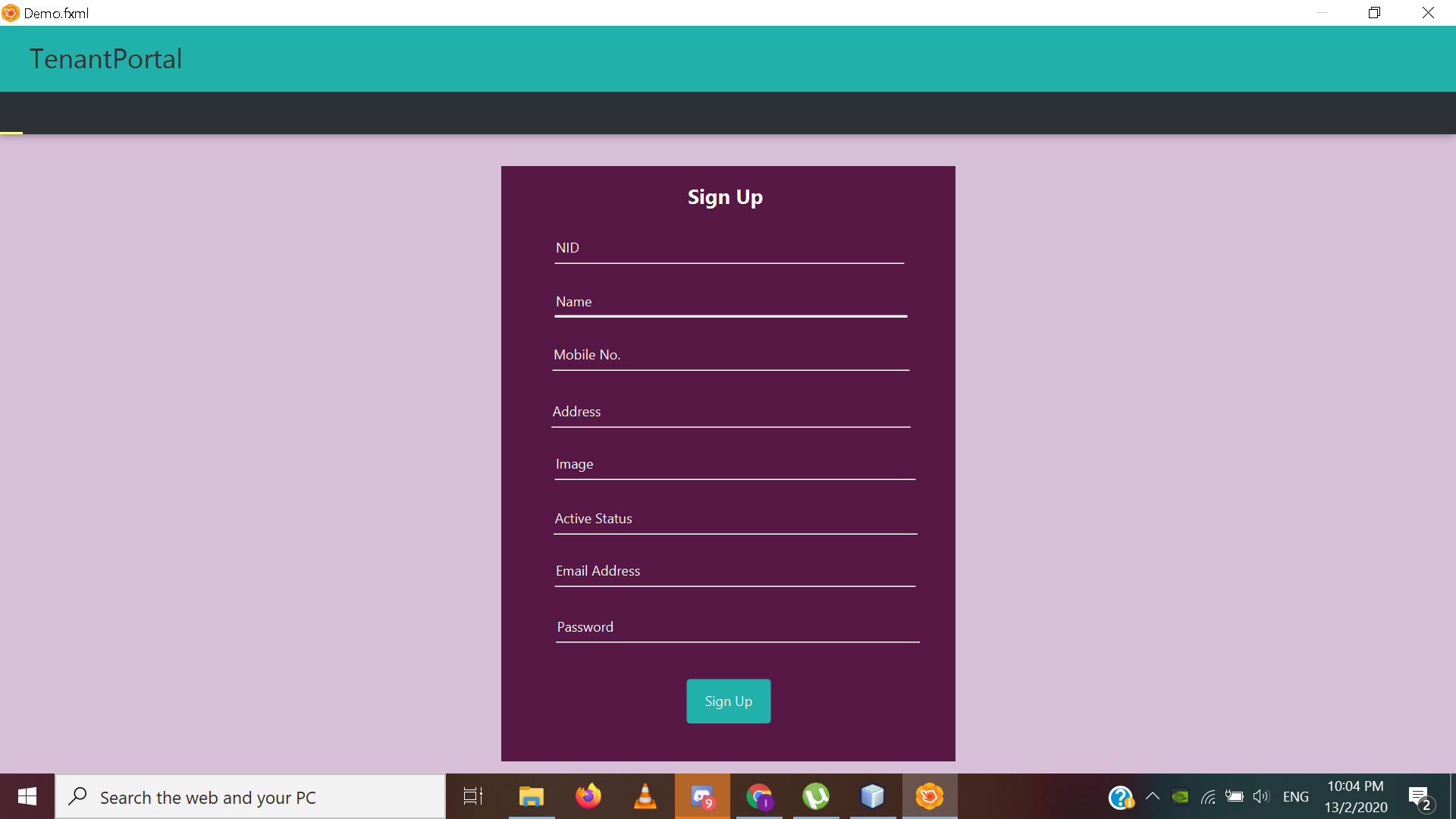
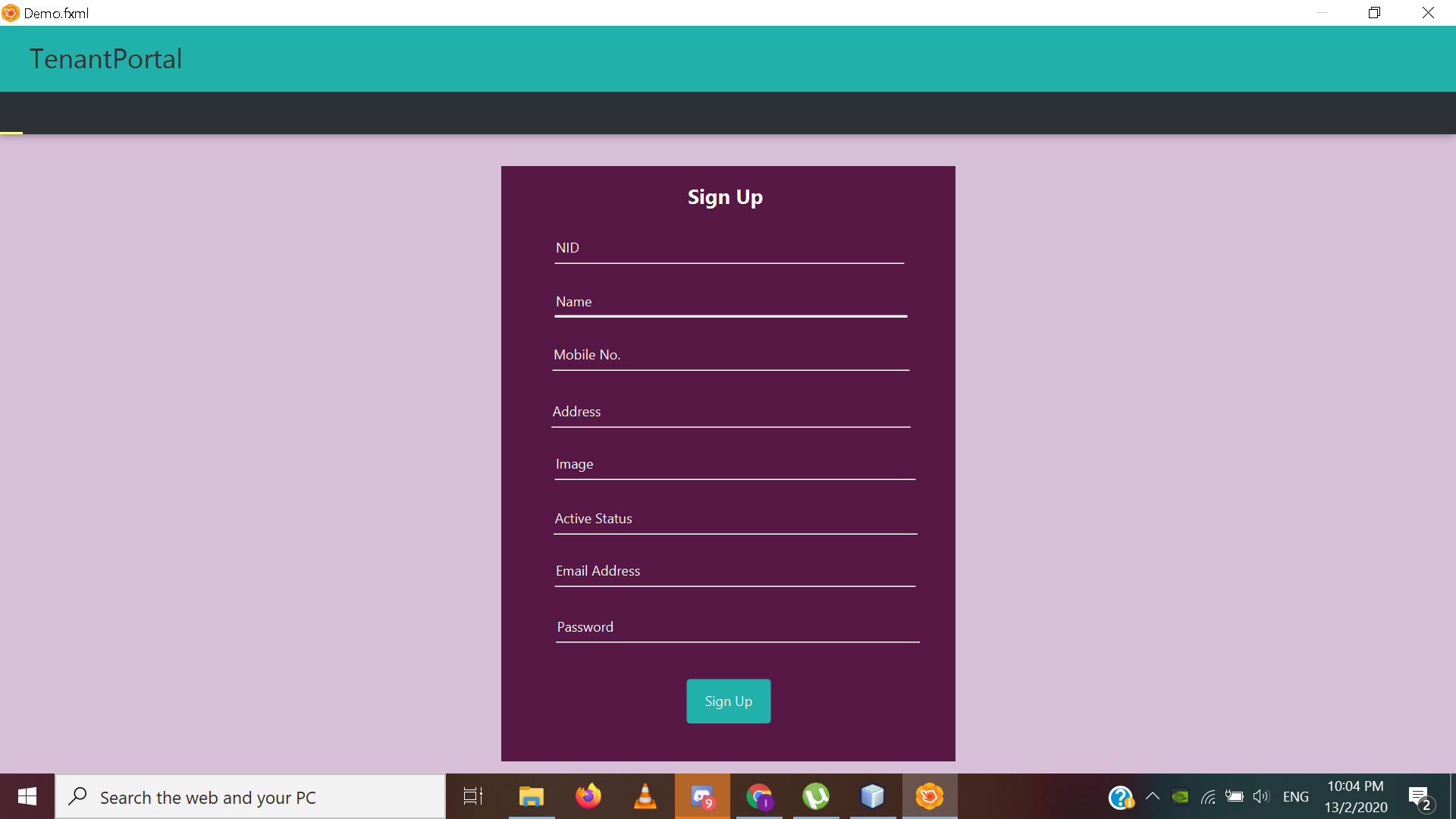
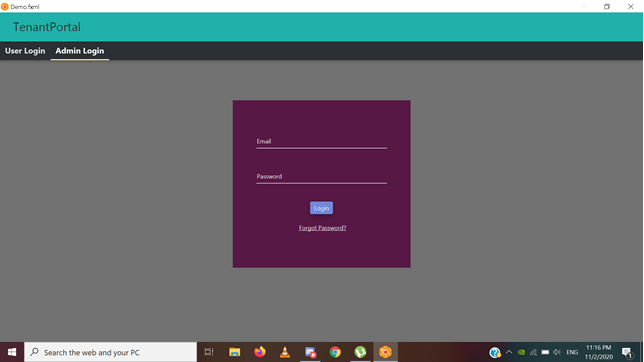
NID: National identity card

FTP: File Transfer Protocol

HTTPS: Hypertext Transfer Protocol Secure

UI: User Interface

# Mockup



# Repository

<https://github.com/Sourav-Debnath/CSE299>