**Dr. Slumlord**

**Or How I Learned to Stop Being Poor and Take Advantage of the Proletariat**

**By:** Alexander Costello, Dexter Elliot, Joshua Escareno, Chase Keller, and Odysseus Valdez

**Overview:** This document lays out the system design specification for the group project assigned for CSCI 2251 during the Summer semester of 2019. It is currently a rough draft and subject to change.

**Context:** The reason for this project’s existence can be attributed to a class assigned group project and is meant to demonstrate several key concepts that we can take away from the class.

The overall purpose of the project (from what I can tell) is to create a web application that can be used to manage rental properties owned by given individuals. With this application a user will be able to log in and view and manage their rental properties. This may mean that they have the application create an invoice for a single month for a single property or for all of the properties owned by a single individual. They will also be able to manage tenants through this application.

**Goals:** The main goal of this project is to create a seamless and easy to use property management system to allow our illustrious slumlords more time to extort the poor and less time worrying about how much they should be squeezing them for! Our most respected clientele cannot be bothered to keep track of how many people they can squeeze into a 2-bedroom apartment so we aim to do that for them all at the click of a button!

Main goals are as follows:

* Create a safe and secure web application that will protect our esteemed client’s user information
* Implement a web ui that will allow our slurmlords to access and edit information about their rental properties and corresponding tenants.
* Create the necessary classes to handle requests from the front end and make calls to the database to perform the necessary state reconciliation transactions
* Display all changed information to the user in a friendly and easy to read manner

**Milestones:** Major milestones are as follows and subject to change (dates and times intentionally omitted) :

* Create working design documentation
* Create and seed database with initial information and a test user with records in all relevant tables
* Create DAOs to handle calls from java to the database
* Create User java classes that handle the back end logic for user function calls made from the web ui
* Create Web UI

We will measure the success of the above milestones through the use of automated testing where applicable. Since I believe we are using HTML5 we will not be running the UI through automated tests.

**Proposed Solution:** Our proposed solution is a straight forward and generic web application, with “client” and server architecture. It seems as though it is a requirement to have a dual client server architecture as we will have a client UI and UI server as well as a backend and DB client/server.

**Secure Failure:** My apologies for this being shoe-horned in at the end of the document but as it is a requirement we feel it is necessary to inform our generous and respectable landlords that in the event that the software fails in any form it will do so securely. This can be addressed in a 3 pronged approach:

* If a user cannot be validated the software will not allow the user to access private information of any individual stored within the system
* If the connection to the database is lost in the middle of a transaction, data changes will be rolled back and not committed which will reduce the chance of corrupted data entering the database
* If the web services cannot be reached the webpage will display a 401 or a 503 error depending on the circumstances and if the database is unavailable users will be able to make changes in the client while a connection to the database attempts to be established.