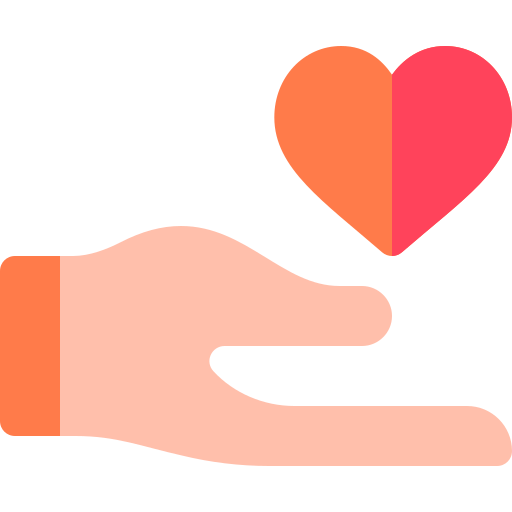


Algonquin Homes

Systems Design and   
Project Management

Prepared by The Doctors 

Alekhya Tamarapalli  
alekhyat05@gmail.com

### Table of Contents

[**Table of Contents**](#_d14eskk26lim) **2**

[**1. Systems Design**](#_dobl27nh8wey) **3**

[1.1 Environment Description](#_hs7lfwgea0iu) 3

[1.2 Application Components Description](#_wpf6qjru5laj) 4

[1.3 User Interface and User Experience Design](#_wi78nw1fnb21) 6

[Figure 1.3.1 Create Buyer Profile](#_e40bxq1zeijl) 7

[Figure 1.3.2 Create Purchase Agreement](#_tjzgcceu21f0) 7

[1.4 Database Design](#_n92i5f72g7cp) 9

[Figure 1.4.1 Draft Database Schema](#_6oo24c707h04) 9

[Figure 1.4.2 All Basic Relationships](#_820drk8wrk4e) 10

[Figure 1.4.3 Purchase Agreement Relationships](#_cfn426k9suno) 11

[1.5 System Integrity, Security and Privacy](#_5o4u4v7f7uap) 12

[1.5.1 System Integrity](#_ixhowsu0g49t) 12

[1.5.2 System Security](#_6npmzuik69ay) 13

[1.5.3 System Privacy](#_h1d07segkj5q) 13

[**2. Project Management Plan**](#_rcpgf5v9ohg3) **14**

[2.1 The Agile Methodology](#_ojn54naqydwn) 14

[Figure 2.1.1 Sample Product Backlog](#_yxkjiaeuove8) 15

## 1. Systems Design

### 1.1 Environment Description

Algonquin homes currently use manual sales systems to monitor their customer data and update the purchase information. Our goal is to develop a web application that is user friendly for sales agents and customers considering specifications for Algonquin homes.

The web application will consist of two parts: back-end and front-end. Back-end part is going to be deployed on an AWS EC2 instance with an AWS RDS database. Assuming that the load will not be high, an EC2 instance is not going to cost a lot. However, if Algonquin Homes would like, it is possible to deploy the application on their own web-server. All communications will be done over HTTPS protocol that will allow us to protect all sensitive data.

User interface will allow employees to enter information and needs. Also, managers can create profiles for customers. Available slots or houses which are already stored and updated in the database will be shown to the user. A customer would have an option to choose the viewing date according to the sales agent schedule that is already being processed and updated on a regular basis and it is visible. Customers and sales agents would also have the option of updating the data from their end.

There is also involvement of other stakeholders like banks, legal teams, etc, who would work on the background check on clients. They would have access to client information under legal contract.

Since there is a large amount of personal information, there are access levels in the application.

There are different access levels for customers, agents, managers, and administrators. It means that, for example, employees could access only limited data and do not have permission to perform manager tasks.

System may eliminate customer details or profiles that are unused for more than 3 months before establishing, updating and removing the customer profile or information was the sales agent's task. The database also will be modified or disabled depending on the availability of the property.

### 1.2 Application Components Description

Application for Algonquin homes is a combination of both open source and licensed technologies. Since the back-end and front-end parts will be developing separately, there are different stacks of technologies for each part.

Front-end will be a web application written on HTML5, CSS3 (Bootstrap), and JavaScript (jQuery). These days, such a set is very common, convenient, and cheap for development, and there is no reason to change it. It also allows Algonquin Homes to easily maintain and scale the project for its whole life cycle.

The main factors for the view or user interactive is to maintain the page/view load factor which will be faster in this case. Bootstrap and CSS are used for styling the page to make it appealing. It also makes it easy to access and understable. The essential factor for Algonquin Homes is to provide the fields to record the necessary inputs of the clients. The necessary validations have to be effectively implemented. Auto complete fields are also incorporated.

Back-end also consists of the two parts: web-server and database; C# (ASP.NET) and MS SQL Server will be used for each part respectively. These two tools were developed to work with each other. We chose these technologies not only due to the fact that today ASP.NET is used for back-end development ubiquitously but also because our team has professionals in this area. And although there are other open-source server-side options such as Java, Node.js, etc. .NET is preferred over these technologies in this project because of high reliability and trust that is guaranteed by Microsoft.

Back-end will provide REST API for all the available functionality. Front-end will communicate with back-end over HTTPS protocol using JSON for data transfer. The advantage of this choice is that in the future using the same API, a mobile application can be done for the current system.

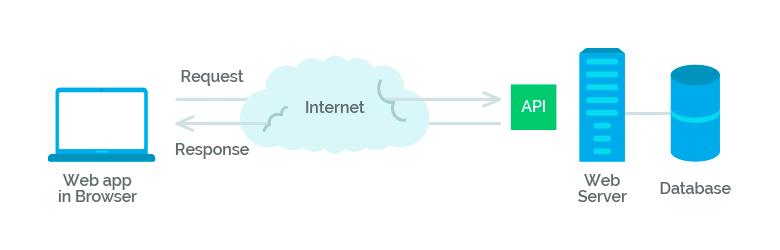
On the back-end, all of the data and functions for each component are placed in different components: in the Model, View, and Controller. MVC pattern states that it has separate components for Model, View to be displayed, and Controller to execute the business logic.

**View:** View provides the API to the front-end, where the website can access and feed the information. This allows the front-end to access the data quickly and dynamically. The API request is performed based on the user-action on the client-side.

**Model:** It contains solely data operations. It does not contain any logic in it. They retrieve and store data from databases. SQL Server database is used to store and dynamically access the data by the user.

**Controller:** Controller is the Major critical part of the system. All the logic is implemented here.

APT.NET Framework is used to implement the logic. To hit the server, the API calls the controller and the corresponding controller function is executed. These APIs and controllers follow the micro-service architecture. Microservice architecture is preferred over a monolithic architecture because microservices provide higher scalability and performance at lower cost. Also, the downtime is negligible during deployment of and feature updates in the server.



### 

### 1.3 User Interface and User Experience Design

Ultimately, research and further understanding of the problem and current situation will inform these decisions throughout the Agile development process. Initially, we have some baseline expectations for the performance and usage of the Home Sales System to be built for Algonquin Homes.

The expectation would be that the Home Sales System would be hosted as a responsive web application online, accessible to users by a web browser and a link. Employees of Algonquin Homes would have access to the Home Sales System by entering a unique login and password combination.

By hosting this application as a responsive web application instead of a native desktop or phone application, this web app can be accessed by users on all of these devices in a format that looks and feels like a native application. By only developing for one format initially, we will save time and effort on the customizations required for each application to run natively on devices.

The perfect technology assumption is the assumption that a system runs under perfect operating and technological conditions. Based on this, the user interfaces and experiences for basic user interactions such as login, logout, password reset, etc. are well established, and will not be included in this report.

The primary user experience and interface revolves around the customer profiles, but more specifically, purchase agreements. In the following two pages, we have outlined the wireframes for two user stories, Create Buyer Profile (Figure 1.3.1), and Create Purchase Agreement (Figure 1.3.2).

### 

### 

#### Figure 1.3.1 Create Buyer Profile

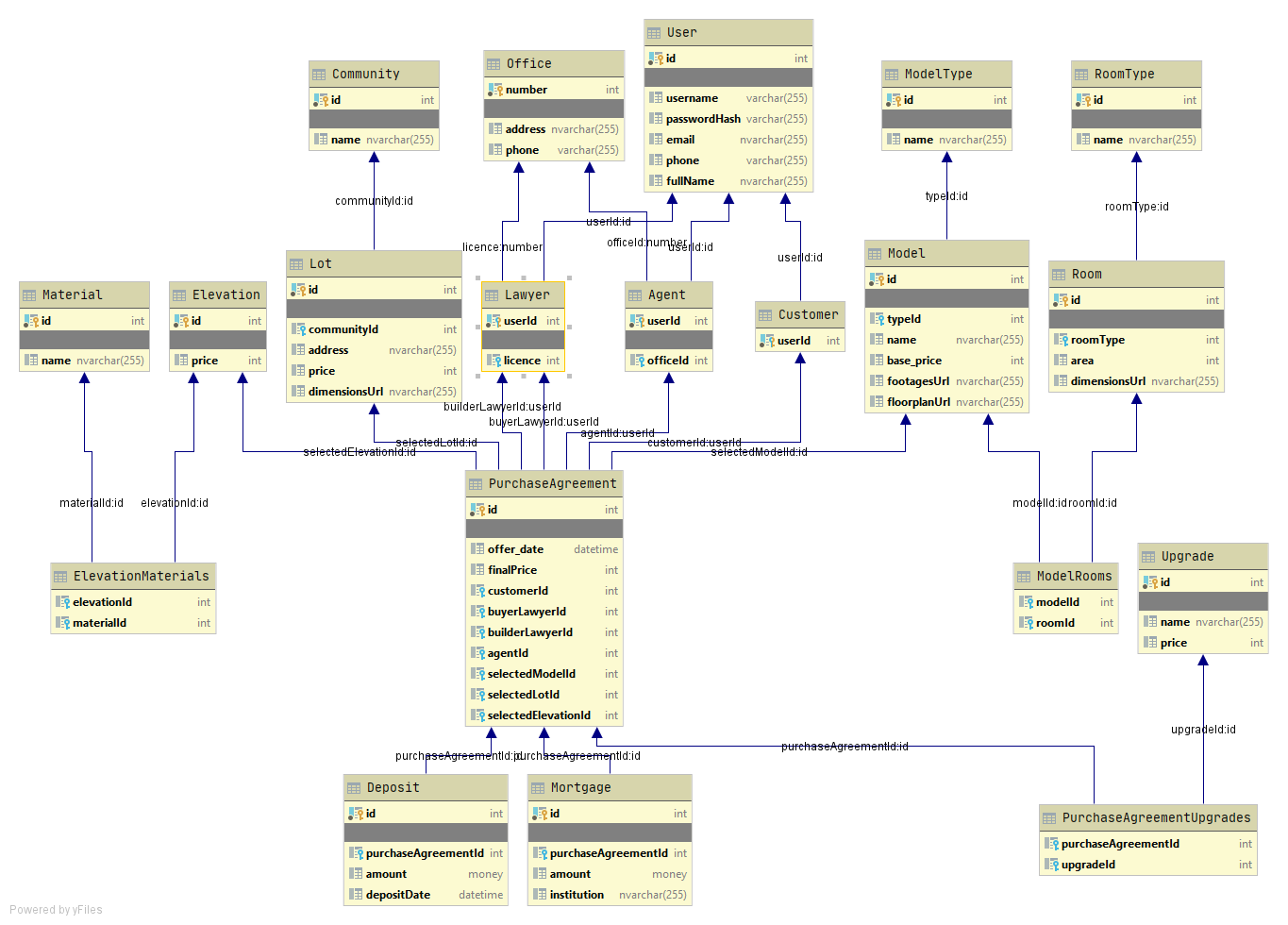
#### Figure 1.3.2 Create Purchase Agreement

### 1.4 Database Design

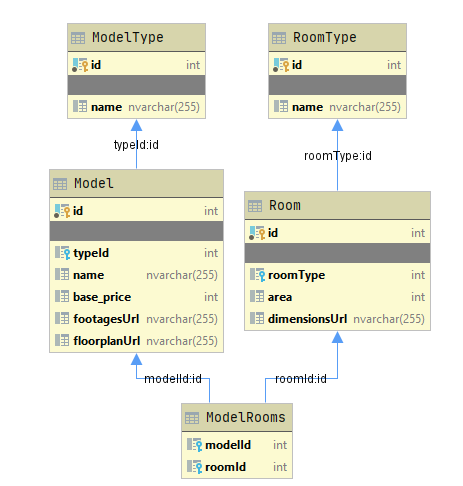
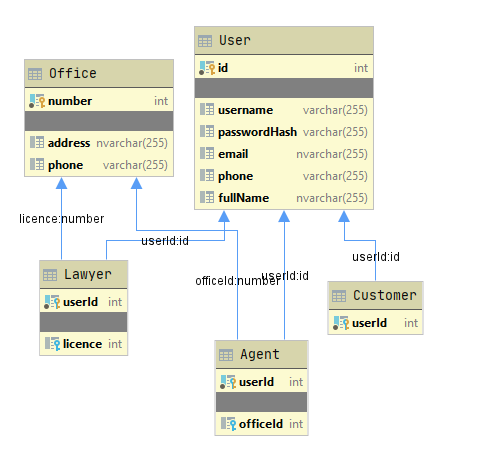
In this case, we have a large amount of relational data and do not have many users because, in general, our application is more for managers, not for customers. It means that for our purposes relational databases will suit better than No-SQL ones. Moreover, since the backend part will be using C# as the main programming language, MS SQL Server is the best choice for the database. It is well compatible with the C# environment. In addition, SQL Server has established itself as one of the most popular databases with high-quality support. Microsoft SQL Server is Microsoft’s flagship database. It is a high-performance database that optimizes all work with the data.

All the data is represented as a set of tables. Some tables are simple, for example, the table “Upgrade” has an ID (primary key), a description, and a price. Other tables are more complicated and have many foreign keys. For instance, columns in the “PurchaseAgreements” table mostly would be foreign keys to the other tables: IDs of the selected upgrades, the ID of the selected model/elevation/community, and so on. The database schema for the application must be designed according to the third normal form design approach. The draft for the database schema is presented below (this is **not** the final version):

#### Figure 1.4.1 Draft Database Schema



#### Figure 1.4.2 All Basic Relationships

****

### 

#### Figure 1.4.3 Purchase Agreement Relationships

### 

### 1.5 System Integrity, Security and Privacy

#### 1.5.1 System Integrity

File integrity is monitored at the point of sale (POS) systems for ensuring that changes to the systems do not allow a breach of mastercard and other customer data. POS systems require file integrity monitoring because they're directly related to the processing of credit cards. Changes to the software system and applications on point of sale (POS) systems can cause system downtime and worse, a breach of buyer data.

Writing log files allows us to maintain an insight to changes or updates performed by a sales representative that are occurring on the server. By recognizing which representative made a change, what exactly changed, and “when” it changed.The sales representative can access the updated profile of the user including all the new payments or details involving the plot and upgrades.

The following steps are made to avoid any issues to the Algonquin Homes Inc (legal or financial issues) and these can be carried out beyond the office hours automatically by the system:

* Log – Changes are logged and alerted.
* Update – A backup of the stored file is taken and stored as changes occur. This allows for changes to be analyzed and the former user profile to be redeployed at any time with one-click.
* Deny Rights – Denies any access to a file and it does not matter what privilege access the user has, access to a file will not be allowed thus denying reads, changes, deletions or additions.
* Data Input - The system limits on entering and changing data, checking the integrity, which can guarantee safe and proper inputs. This can be done by limiting the controls. When the employee creates a new buyer profile there are certain mandatory data that should be collected from the user like employment status, mortgage documents. However it is made compulsory to add the lot number, upgrades if any to the buyer profile and many such fields are required for the purchase agreement.

Adding to it there are certain conditions and requirements like the date of signing the agreement should be the same as when the first deposit is received. The deadlines should be met for second and third deposits according to the ones set by the system. The date of signing the agreement additionally requires 5 days to be accepted by the builder. After 5 days, the system will not allow you to create any changes without special consideration.

#### 1.5.2 System Security

* Maintaining login control: the sales representative would have login specifications to gain access to the user profile and system . If the system notices inactivity for an hour, the user will be logged out and required to log in again. The access will be limited for the employees who are authorized to read, update, and generate data but they will not be able to make any changes to the prices of the lot (the last few lots being an exception only after authorisation).
* Restricting access to data in files: After establishing login restrictions, the employee or user will be able to access the information on the system. According to the roles played by the user, the system will permit some users to read files, and provides other users permission to alter or delete some files, for example, the lawyer and the builders can also register to the system but they are allowed to only read the data. Whereas the other authorised users will be allowed to make approved changes to the profile like making changes to the purchase agreement or changing the price of the last few lots. The changes will be made with approved signatures from the buyer and the other entities involved.
* File encryption: The file can be made unreadable by the unauthorised users by placing a sensitive file into an inaccessible directory.

#### 1.5.3 System Privacy

To ensure that the Information within the framework will be utilized for the allotted reason only, a few steps are taken . The system would include a privacy agreement that the buyer accepts, allowing the system to use their information to proceed with the purchase of homes. Since the system contains information that's profoundly delicate, each client or employee will be completely aware of how to oversee the information within the framework safely and only authorised personnel will be accessing such delicate information. The employees will consent to an agreement that the data will not be used for personal purposes and abide by it.

## 2. Project Management Plan

### 2.1 The Agile Methodology

For this particular project, The Doctors would use the Agile Methodology to plan and manage this project with Algonquin Homes. The Agile Methodologies, hereon to be referred to as simply Agile, defines how this project will be executed, monitored, and controlled throughout its life cycle.

Agile methodology involves stakeholders throughout the inception of the project until the project’s deployment. However, it also keeps the focus on software development considering the needs of Algonquin Homes and customer requirements.

The defining characteristic regarding Agile is that the project will produce and deliver working increments of software in work periods called Sprints, which are typically defined as a two to four week period. Agile breaks down the problem into smaller chunks, referred to as User Stories. With a rough outline of the expected Use Cases and User Stories, the project development team is able to make regular progress towards the development of the project as a whole by focusing on individual experiences and activities required by the system.

Each sprint will focus on a particular use case or set of User Stories and deliver working increments of software at the end of each Sprint. These working increments of software are reviewed in an iterative process which enables our team to routinely evaluate and address the requirements of the project. Agile is the most successful when all stakeholders are engaged throughout the development of the project and individual User Stories to ensure that work is in-line with client needs and project expectations.

The fast pace of change that we experience in business and technology today is a primary reason why Agile project management is useful to ensure the project delivered meets the needs of Algonquin Homes and their customers. The value of delivering tangible software increments allows intermittent testing, refinement, and iterations that ultimately have the ability to develop a product that truly meets the needs of the project.

One member of the scrum team will be the Scrum Master, and lead day-to-day operations by managing a daily stand-up meeting with all members of the scrum team. During this short stand-up meeting, each team member will review their list of “Doing, Done, Next and Blocked”. Doing is anything they are currently working in progress, Done is something completed the day prior, Next is what backlog item will come next after completing the previous task, and Blocked means anything the team member is stuck with and requires additional support. It is the responsibility of the Scrum Master to remove these blocks and enable the scrum team to produce bi-weekly or monthly increments of working software according to the Sprint backlog.

#### Figure 2.1.1 Sample Product Backlog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Priority** | **Title** | **Description** | **Acceptance Criteria** | **Sprint Backlog** |
| User Stories | | | | | |
| 1 | 1 | CRUD Buyer Profile | As an employee of Algonquin Homes, I need the ability to create, read, update, and delete Buyer profiles. | * Back-end: There are endpoints for each operation in API that make appropriate changes in the DB; * Front-end: There is a web page that allows user to perform all the CRUD operations; * Unit tests for each scenario are passed. | 1 |
| 2 | 1 | CRUD Purchase Agreement | As an employee of Algonquin Homes, I need the ability to create, read, update, and delete Purchase Agreements | * Back-end: There are endpoints for each operation in API that make appropriate changes in the DB; * Front-end: There is a web page that allows user to perform all the CRUD operations; * Unit tests for each scenario are passed. | 1 |
| 3 | 1 | CRUD Lawyer Profile | As an employee of Algonquin Homes, I need the ability to create, read, update, and delete Lawyer Profiles | * Back-end: There are endpoints for each operation in API that make appropriate changes in the DB; * Front-end: There is a web page that allows user to perform all the CRUD operations; * Unit tests for each scenario are passed. | 1 |
| 4 | 2 | CRUD Model Information | As an employee of Algonquin Homes, I need the ability to create, read, update, and delete Model Home types and their related information. | * Back-end: There are endpoints for each operation in API that make appropriate changes in the DB; * Front-end: There is a web page that allows user to perform all the CRUD operations; * Unit tests for each scenario are passed. | 2 |
| 5 | 2 | CRUD Community | As an employee of Algonquin Homes, I need the ability to create, read, update, and delete Communities and their related information. | * Back-end: There are endpoints for each operation in API that make appropriate changes in the DB; * Front-end: There is a web page that allows user to perform all the CRUD operations; * Unit tests for each scenario are passed. | 2 |