

**DOCUMENTATION**

***Of project***

**ONLINE HELP DESK SYSTEM**

**Group members :** Tran Khac Tuan (Leader)

Nguyen Thi Phuong Nam

Vo Van Hieu

**Instructor : *Mr. Nguyen Khanh***

**Batch : 150**

**Group : 5**

**Semester : 3**

***Da Nang, June 2020***

**SOFTECH APTECH**

**PROJECT SEMESTER 3**

**ONLINE HELP DESK SYSTEM**

***Da Nang, June 2020***

**TABLE OF CONTENTS**

[1. Acknowledgement 5](#_Toc46628201)

[2. Project Plane 5](#_Toc46628202)

[2.1 Introduction 5](#_Toc46628203)

[2.1.1 Objectives and concentration 6](#_Toc46628204)

[2.1.2 Motivation 6](#_Toc46628205)

[2.1.3 Scope and limitations 7](#_Toc46628206)

[2.2 Project Organization 7](#_Toc46628207)

[2.3 Schedule 7](#_Toc46628208)

[3. Software Requirements Specification 8](#_Toc46628209)

[3.1 User Requirements Definition: 8](#_Toc46628210)

[3.2 System Architecture: 8](#_Toc46628211)

[3.3 Functional requirements 9](#_Toc46628212)

[3.3.1 Administrator: 9](#_Toc46628213)

[3.3.2 Students: 9](#_Toc46628214)

[3.3.3 Facility-heads: 9](#_Toc46628215)

[3.3.4 Assignor: 10](#_Toc46628216)

[3.4 Nonfunctional requirements 10](#_Toc46628217)

[3.4.1 Portability requirements: 10](#_Toc46628218)

[3.4.2 Usability requirements: 10](#_Toc46628219)

[3.4.3 Space requirements: 10](#_Toc46628220)

[3.5 System Requirement Specification: 10](#_Toc46628221)

[3.5.1 Hardware Requirements 10](#_Toc46628222)

[3.5.2 Software Requirements 10](#_Toc46628223)

[4. System Design: 11](#_Toc46628224)

[4.1 Context Diagram 11](#_Toc46628225)

[4.2 Modelling 11](#_Toc46628226)

[4.2.1 Use case diagram 12](#_Toc46628227)

[4.2.2 Activity diagram 18](#_Toc46628228)

[4.2.3 Entity Relationship Diagram (Template) 25](#_Toc46628229)

[4.3 Database structure 26](#_Toc46628230)

[4.4 Screen shot 26](#_Toc46628231)

[4.4.1 Login page 26](#_Toc46628232)

[4.4.2 Home page 27](#_Toc46628233)

[4.4.3 View users in role Page 27](#_Toc46628234)

[4.4.4 User profile page 28](#_Toc46628235)

[5. Source code management 29](#_Toc46628236)

[6. Testing 30](#_Toc46628237)

[6.1 Purpose of Testing 30](#_Toc46628238)

[6.2 Non-functional Testing 31](#_Toc46628239)

[6.2.1 Reliability testing 31](#_Toc46628240)

[6.2.2 Usability testing 31](#_Toc46628241)

[6.2.3 Performance testing 32](#_Toc46628242)

[6.2.4 Security testing 32](#_Toc46628243)

[7. Deployment 33](#_Toc46628244)

[7.1 Deployment Goals 33](#_Toc46628245)

[7.2 Why on cloud? 33](#_Toc46628246)

[7.3 Publishing steps: 34](#_Toc46628247)

[7.3.1 Export Profile of server and Database information on Azure 34](#_Toc46628248)

[7.3.2 Open project & Import publish settings 34](#_Toc46628249)

[7.3.3 Validates Connection: 36](#_Toc46628250)

[7.3.4 Paste the Connection string 36](#_Toc46628251)

[7.3.5 Publish page 37](#_Toc46628252)

[7.3.6 Start publish 37](#_Toc46628253)

[7.3.7 Login page at first sight 38](#_Toc46628254)

[8. Conclusions 39](#_Toc46628255)

# Acknowledgement

It is our proud privilege to release the feelings of our gratitude to several persons who helped us directly or indirectly to conduct this research project work - Online Help Desk.

We express our deep sense of gratitude to our teacher and instructor Mr. Nguyen Khanh, teacher of International Programmer Training Center Softech Aptech for the thoughtfulness, sincere encouragement in completing this project.

We would like to thank all of our friends who have more or less contributed to the preparation of this project report and will be always indebted to them.

We are also extremely thankful to Softech-Aptech for creating the opportunity for us to successfully implement this project.

Making Online Help Desk System has indeed helped us to explore more knowledgeable avenues related to our topic and sure, it will help us in our future.

# Project Plane

## Introduction

This document will propose all features and procedures to develop the system.

This document specially containing details about objectives, scope limitation, process model, primary requirements, team development, possible project risks, project schedule, and finally monitoring and reporting mechanisms.

This project is aimed at developing an Online Help Desk (OHD) for the facilities in the campus. This is an Intranet based application that can be accessed throughout the campus. This system can be used to automate the workflow of service requests for the various facilities in the campus. This is one integrated system that covers different kinds of facilities like class-rooms, labs, hostels, mess, canteen, gymnasium, computer centre, faculty club etc. Registered users (students, faculty, lab-assistants and others) will be able to log in a request for service for any of the supported facilities. These requests will be sent to the concerned people, who are also valid users of the system, to get them resolved. There are features like email notifications/reminders, addition of a new facility to the system, report generators… in this system.

Identification of the student requirements and drawbacks of the same kind systems leads to be the design of the management system compatible with reality. We can improve system efficiency without making bad influence to user experience.

### Objectives and concentration

* System to manage problem about facilities in the campus, can be used to automate the workflow of service requests for the various facilities in the campus.
* Easy way for students to send request, the assignor to assign and the facility-head to resolve the problem.
* Allows to store requests.
* Admin manage all person in system (student, assignor, facility-head).

### Motivation

Online Help Desk (OHD) systems are designed to manage, store, and solve problems about facilities on campus. By different groups of people such as: students, faculty, lab-assistants and others will be able to access controlled information systems and distribute information flexibly. The goal of cooperation is: approach-solve problems faster and better by applying their common understanding, choosing the most appropriate problem-solvers. Using OHD help improves productivity, speeds up result-making and optimizes of making a right-decisions, it also helps to intercept precious intellectual fortune and time. To prove such kind of improvement to productivity and to make easier our everyday working life, it was needed from the universities to make an inside system for OHD management. Specifically, it is difficult to find the right support person to hand over processing requests to avoid wasting valuable time searching and sending, describing new requests of students. So OHD system help maintain an efficient system of collection, sorting and delivery of users’ information.

The system is essential because it helps students send requests without knowing who to send, helps assignor assign requests to facility-head without knowing them. When they first arrive at the school, the student does not know who the recipient of the problem is, the assignor does not know who to send the request to, so a list of information is essential. An OHD system is useful while starting in new school to save the time wasted with manual method of checking users’ information.

For reduce consuming necessary time, keeping all troubles less, and to organize all the requirements for lab-assistants and most importantly monitor the requirements that need to be addressed or keep an eye out for errors which occurred during the assignment process. So, a good web OHD system was under consideration to consider everyday use and needs, the aim was to make an inside system for the university. This system is to help students report facilities' faults for management in the fastest and most convenient way, assignor selects the most appropriate processor through the information stored in the database. without effort, helps facility-head to reach and solve problems as quickly as possible. OHD system is needed, for helping to organize and keep an eye on the requests and its process. The most important part is, the system has a request tracking system that can remove requests that are not practical or unsuitable for resolution without going to the site for review

### Scope and limitations

* Design only for facilities in the campus.
* Only works when connected to the internet network.

## Project Organization

|  |  |  |
| --- | --- | --- |
|  | **Job Title** | **Description** |
| 1 | Project Manager | * Manages all processes in the project. * Describe how the software should work. * Assigns tasks for suitable people. |
| 2 | SW Analyst | * Analyze the business requirements for the Clinic system. |
| 3 | Software Designer | * Designs the system models and diagrams. * Decide and implements a suitable framework or technique. * Outline required tasks to PM. |
| 4 | Database Administrator | * Design database. * Perform SQL. * Validate input data. |
| 5 | Developers | * Do everything required by superior. * Optimize code. |

## Schedule

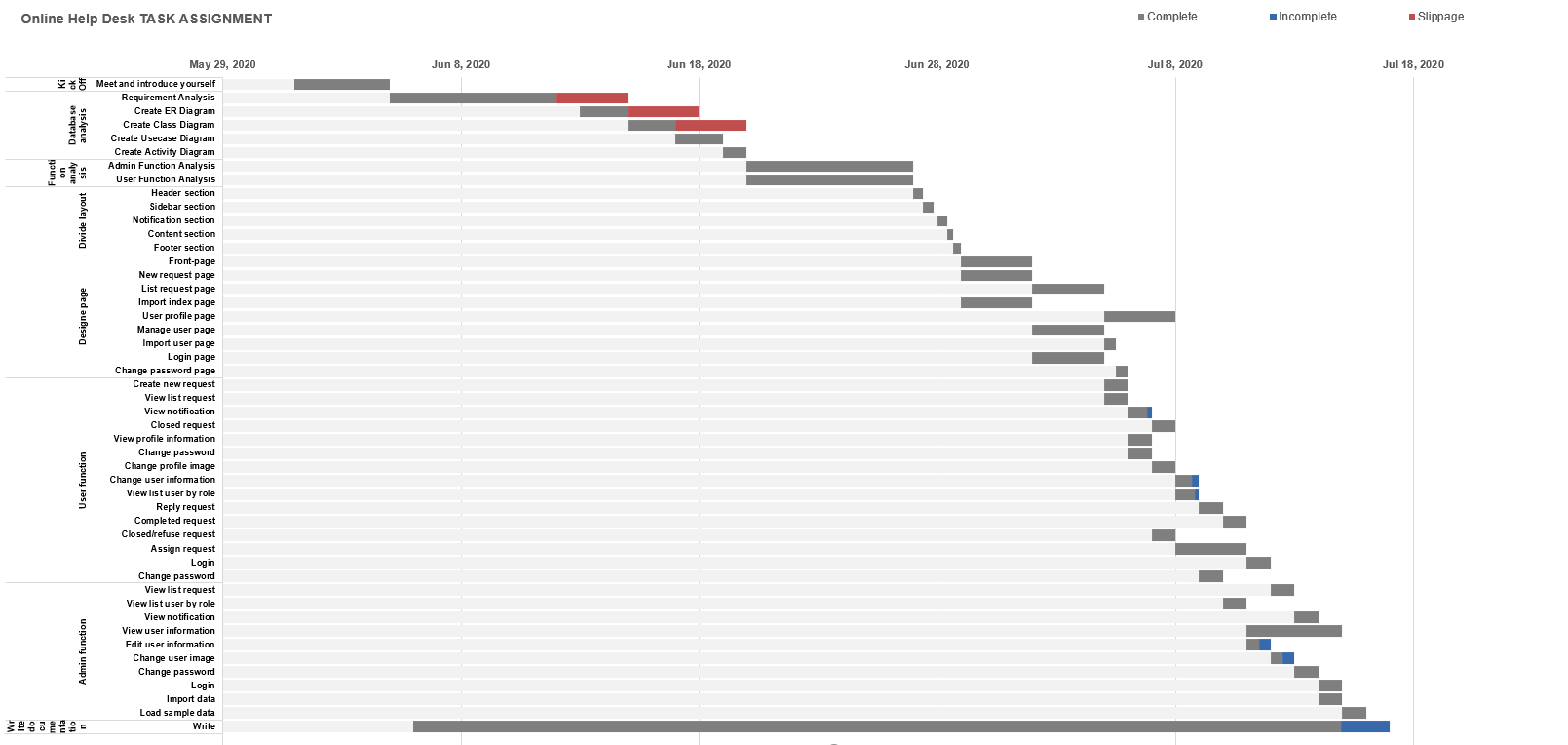


Figure 1. Schedule

# Software Requirements Specification

Taking into account the comparative analysis stated in the previous section we could start specifying the requirements that our website should achieve. As a basis, an article on all the different requirements for software development was taken into account during this process.

## User Requirements Definition:

The user requirement for this system is to make the system fast, flexible, less prone to error, reduce expenses and save the time.

* Less human error.
* Strength and strain of manual labor can be reduced.
* High security.
* Data redundancy can be avoided to some extent.
* Easy to use.
* Easy data updating.
* Easy record keeping.
* Backup data can be easily generated.

## System Architecture:

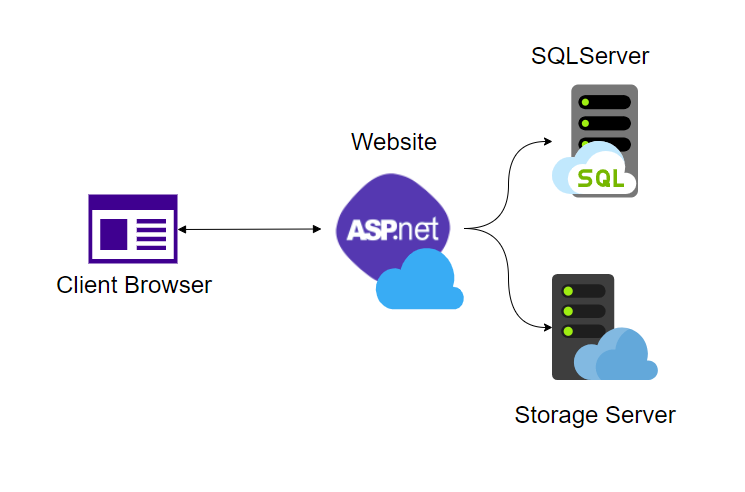


Figure 2. System Architecture

## Functional requirements

* The user should be able to login and manage his/her requests online at any time.
* Database has to store all the information efficiently without any information loss.
* The user can change his profile info at any time.
* Admin can manage all account users (facility-heads, assignors, students), facilities, equipment...

### Administrator:

* Login to the first page
* Manage his/her information detail
* Change the password after login into the system
* Create/ Manage account of: students, facility-heads, assignors
* Create/ Manage facilities, equipment
* View all requests
* Check all status requests
* View notifications
* Logout

### Students:

* Login to the first page
* Manage his/her information detail
* Change the password after login into the system
* Create requests in the system
* Check status of the request created by him/her
* View created requests
* Edit/ Delete requests created by him/her
* View notifications
* Logout

### Facility-heads:

* Login to the first page
* Manage his/her information detail
* Change the password after login into the system
* View information of: other facility-heads, assignors, students.
* Check status requests
* View all requests
* Manage (Resolve/Delete) requests
* View notifications
* Logout

### Assignor:

* Login to the first page
* Manage his/her information
* Change the password after login into the system
* View information of: other assignors, facility-heads, students.
* View all requests
* View new requests
* Check status requests
* Manage (Assign/Delete) requests
* Assign request to suitable facility-head
* View notifications
* Logout

## Nonfunctional requirements

### Portability requirements:

* A website has to be compatible with different popular web browsers (Google Chrome, Internet Explorer 8+)

### Usability requirements:

* Interface elements (e.g. menus) should be easy to understand.
* The user should be able to learn to use a system in less than 30 minutes.
* Time required for login less than 5 minutes.
* Error messages should explain how to recover from the error.
* Actions which cannot be undone should ask for confirmation.
* Responsive design should be implemented

### Space requirements:

* User needs enough disk space and RAM for web browser

## System Requirement Specification:

### Hardware Requirements

All parts of server included database server base on SQL Server. So, client can use any devices to access and manage.

* PC, smartphone, tablet, etc.

### Software Requirements

* Database: SQL Server
* Webserver: IIS
* Application: Visual Studio
* Programming language: C#

# System Design:

## Context Diagram

This diagram represents what are the bounders and scope of Online Help Desk System project. It describes the main objective of the system and its entities involved.

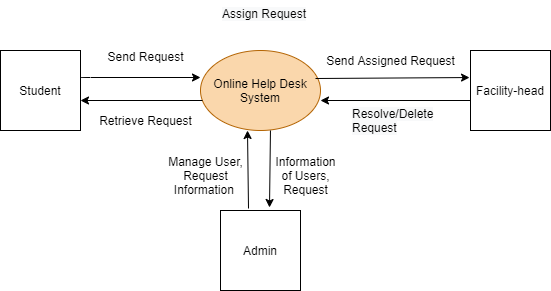


Figure 3. Context Diagram

## Modelling

Modelling is like building a representation of things in the „real world‟ and allowing ideas to be investigated. In fact, model is more likely a way of expressing a particular view of a system. Mainly modelling is used to:

* understand the problems involved in building some system
* an aid to communication between those involved in the project
* a component of the methods used in development activities such as the analysis of the requirements

The way modelling is used in this project is called Unified Modelling Language (UML) that is a standard language for specifying, visualizing, constructing, and documenting the artefacts of systems, as well as for business modelling and other non-software systems. The UML represents a collection of best practices that have proven successful in the modelling of large and complex systems. It is an important part of developing system and their development process. The UML uses mostly graphical notations to express the design of projects, it helps project teams communicate, explore potential designs, and validate the architectural design of the system.

The primary goals in the design of the UML were:

* Provide users with a ready-to-use, expressive visual modelling language so

they can develop and exchange meaningful models.

* Provide extensibility and specialization mechanisms to extend the core concepts.
* Be independent of particular programming languages and development processes.

Provide a formal basis for understanding the modelling language, we use <https://app.diagrams.net/> to achieve it.

### Use case diagram

Use Cases are text stories, widely used to discover and record requirements. Use cases need to be more detailed or structured and emphasize the user goals and perspective. A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a use case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals, and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.

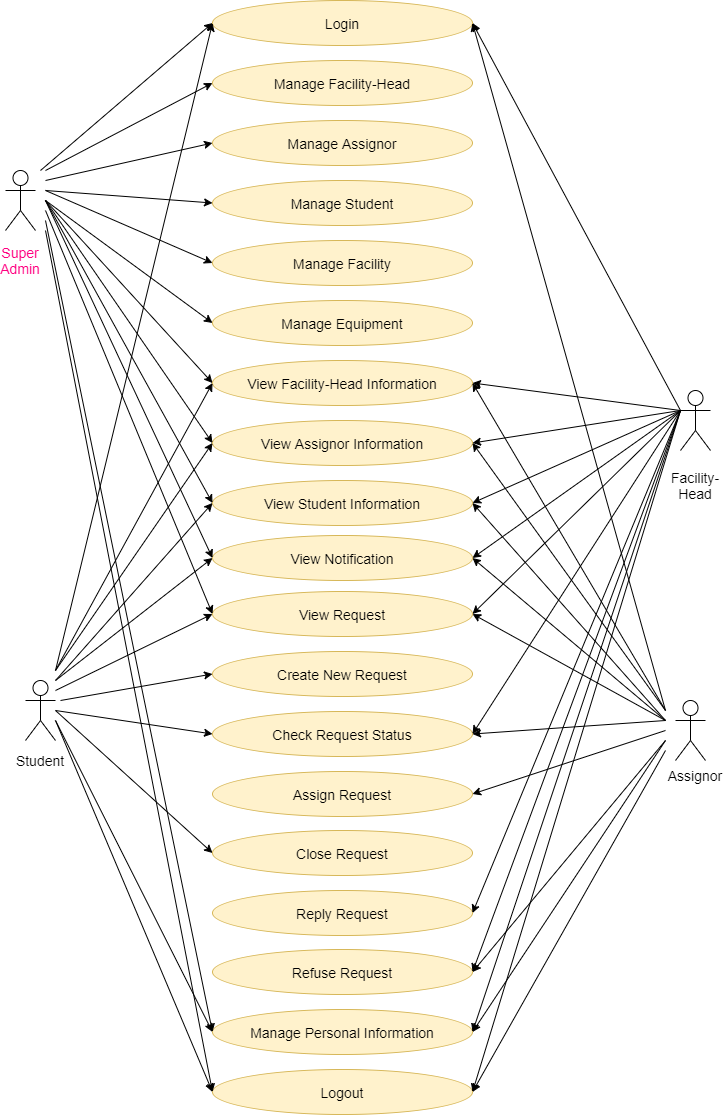


Figure 4. Use case diagram of OHD system (Full)

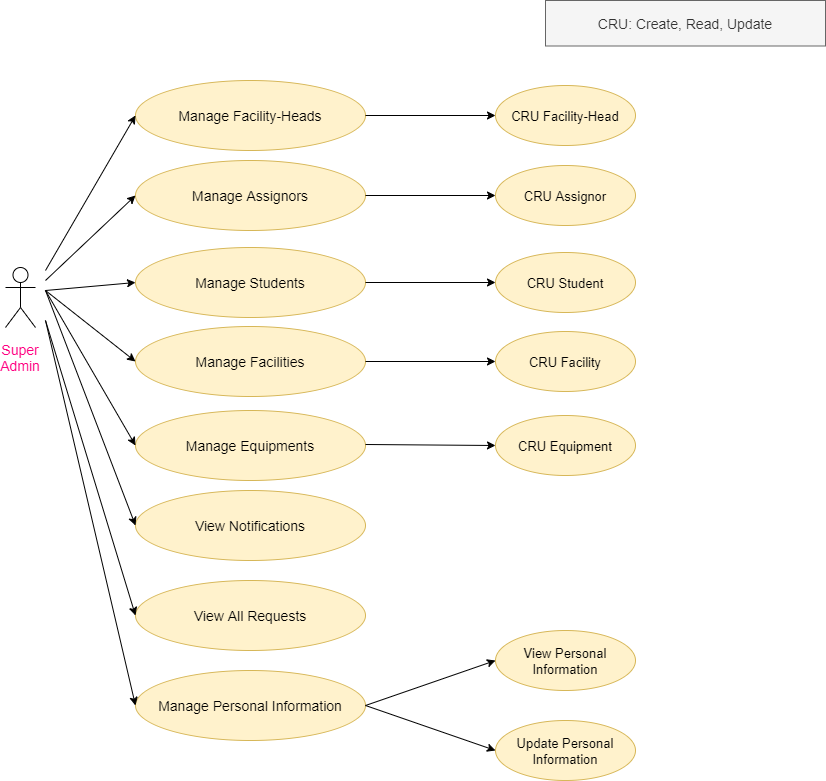


Figure 5. Use case diagram (Admin)

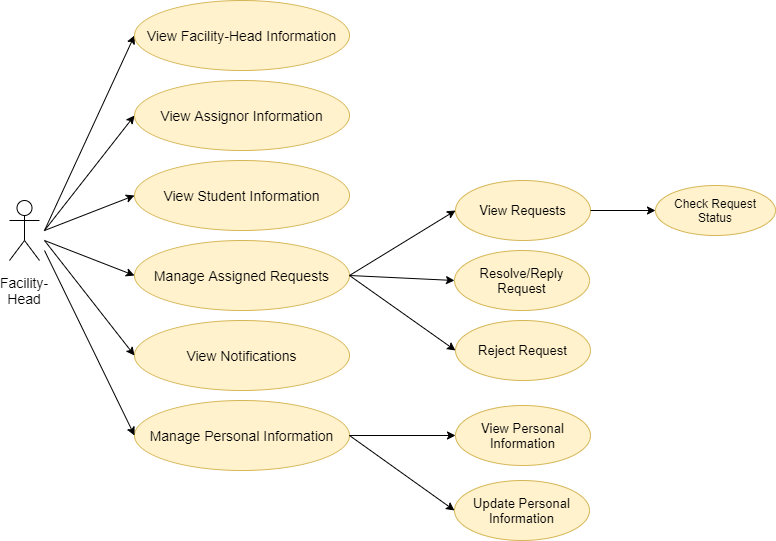


Figure 6. Use case diagram (Facility-head)

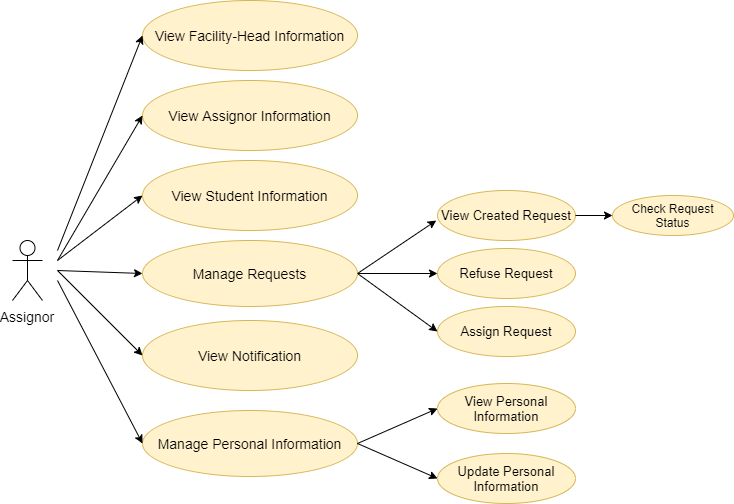


Figure 7. Use case diagram (Assignor)

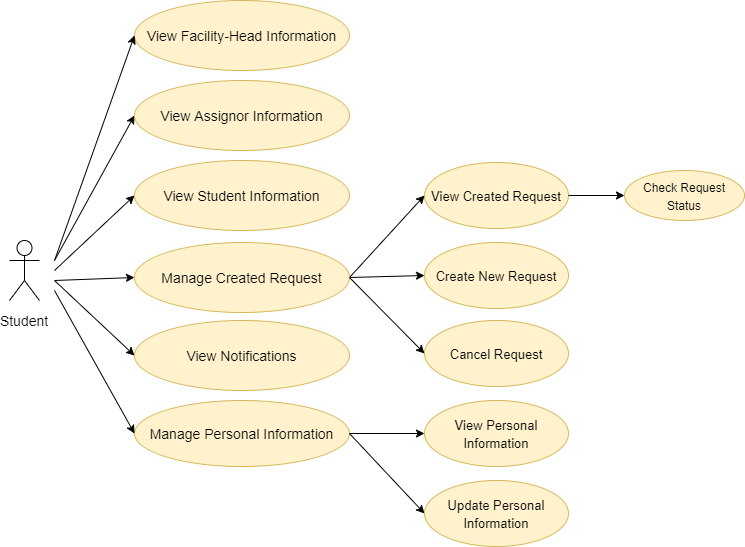


Figure 8. Use case diagram (Student)

### Activity diagram

The basic purpose of activity diagrams is capturing the dynamic behavior of the system. It is used to show the message flow from one object to another, but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as:

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

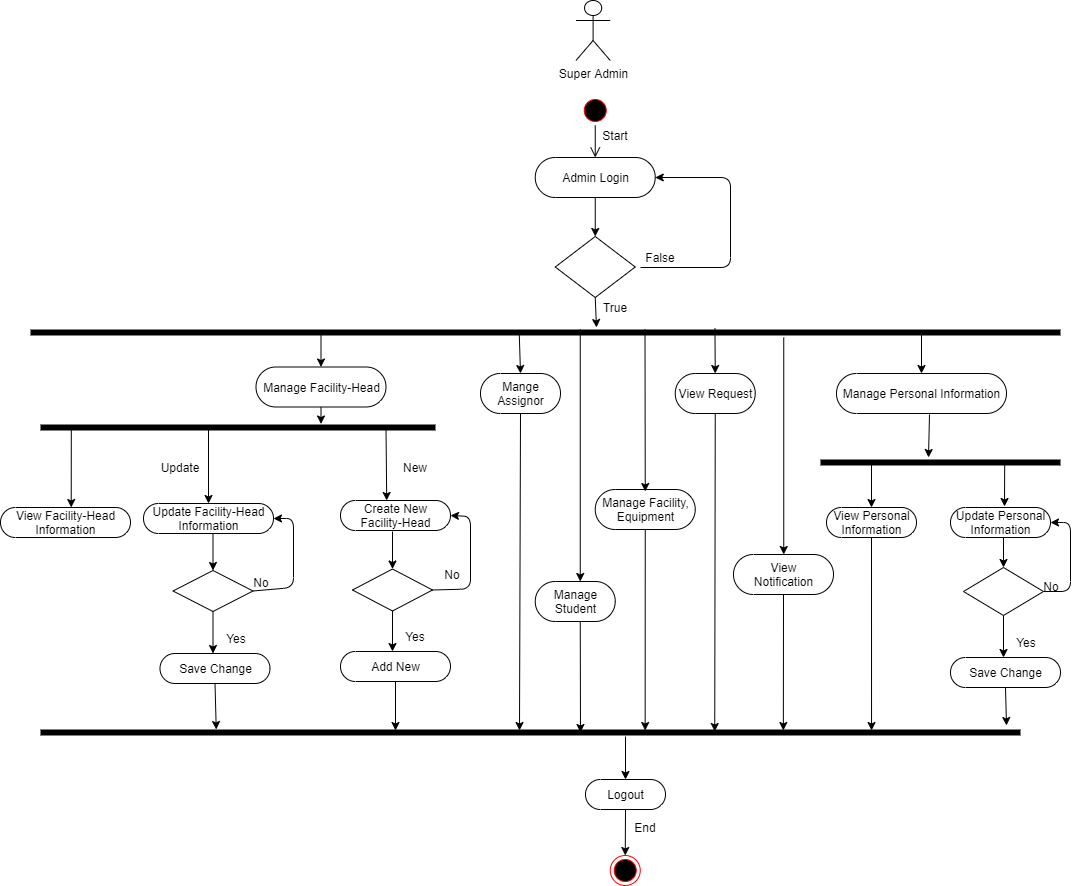


Figure 9. Activity diagram (Super Admin)

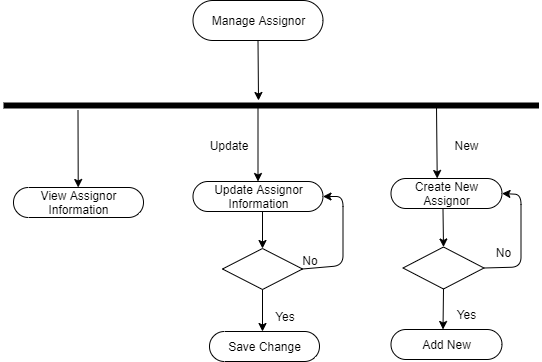


Figure 10. Activity diagram (Manage Assignor)

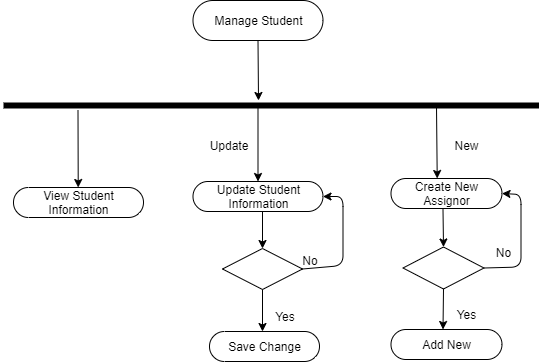


Figure 11. Activity diagram (Manage Student)

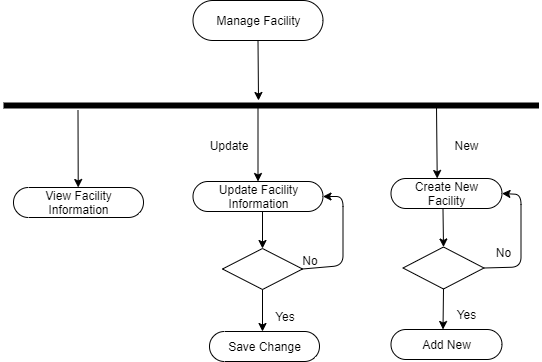


Figure 12. Activity diagram (Manage Facility)

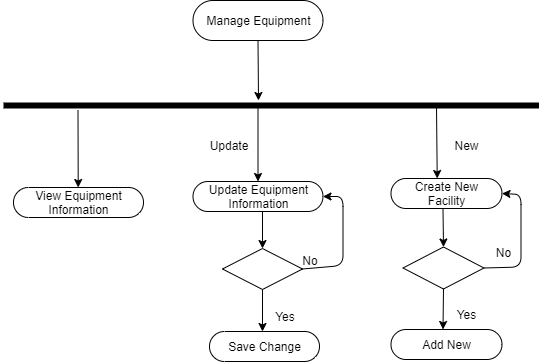


Figure 13. Activity diagram (Manage Equipment)

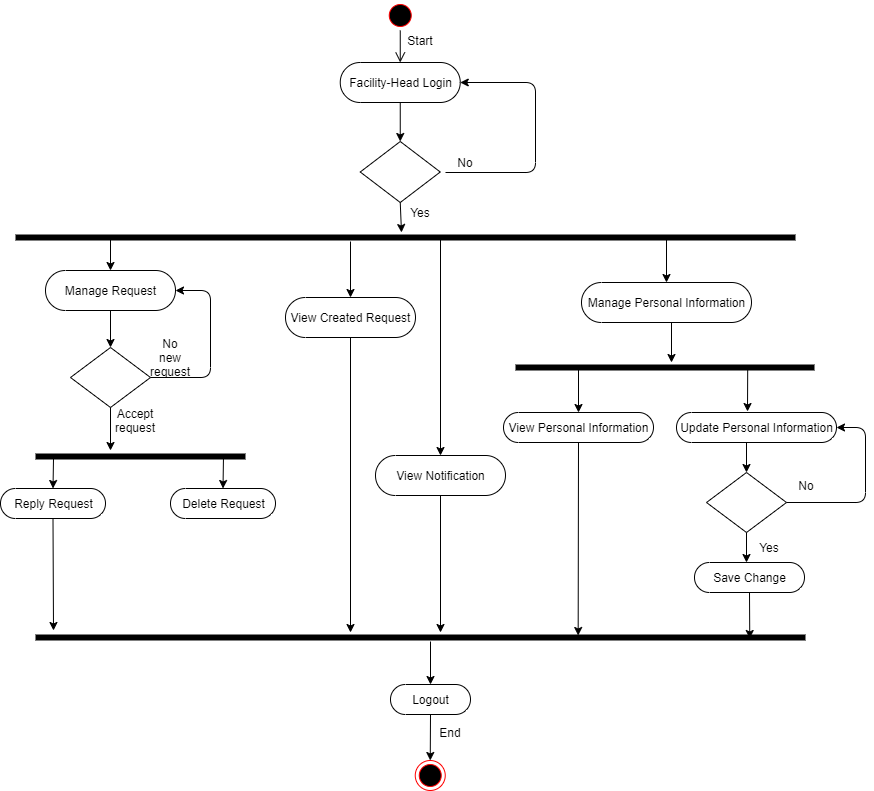


Figure 14. Activity diagram (Facility-head)

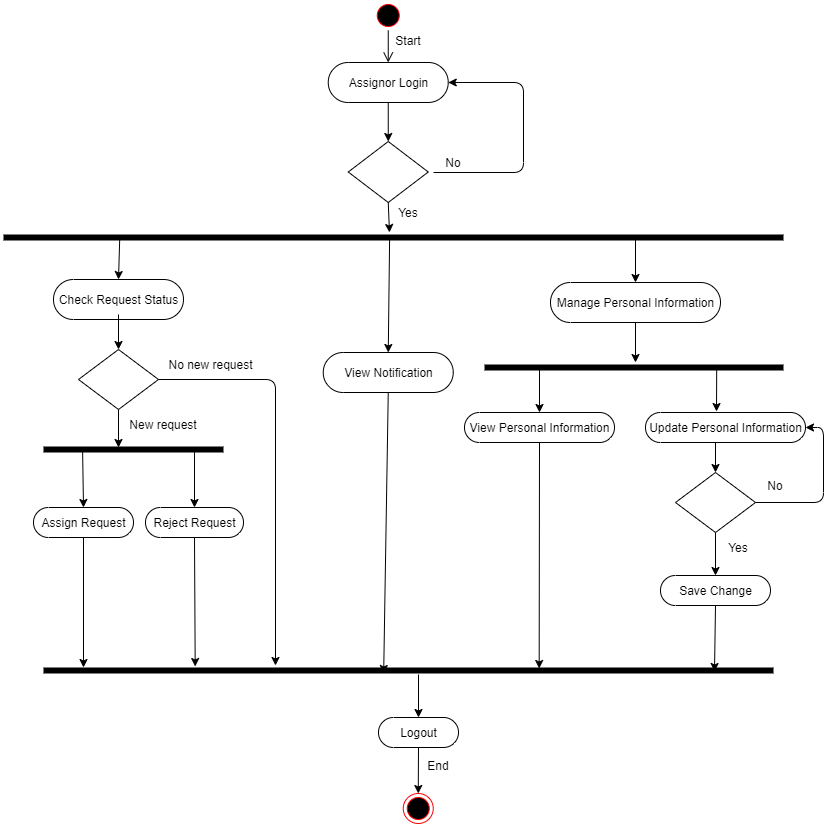


Figure 15. Activity diagram (Assignor)

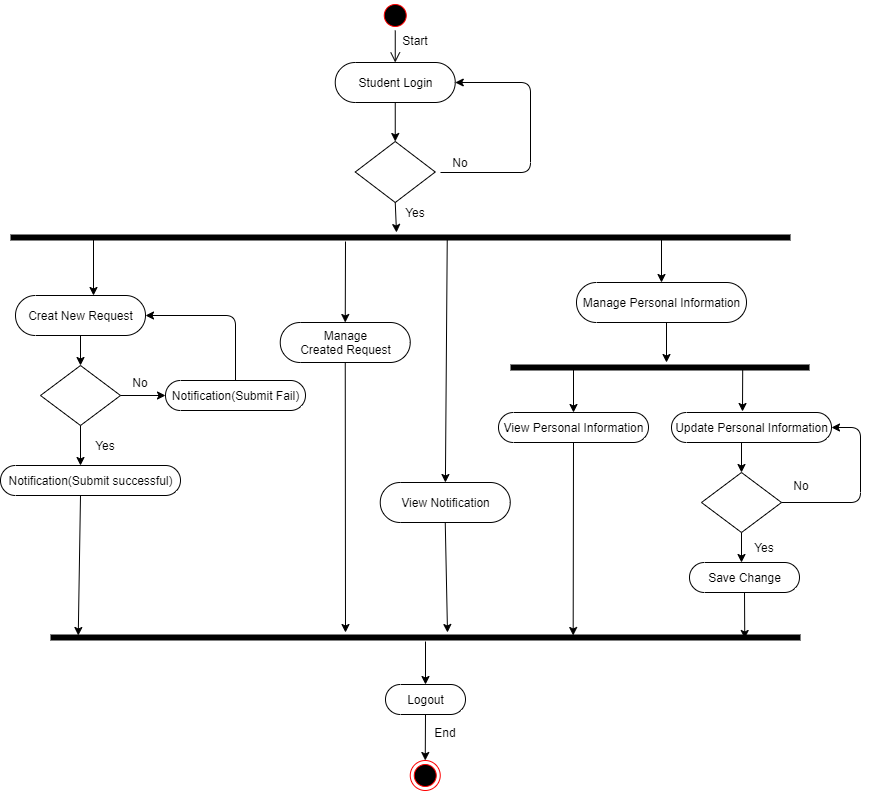


Figure 16. Activity diagram (Student)

### Entity Relationship Diagram (Template)

Creating an Entity Relationship Diagram (ERD) for a proposed database design allows you to see a high-level view of your database before you actually get to creating it, allowing you to make sure you’re capturing everything you need to capture in your various tables and make tweaks before you actually create the database (when it will be much more difficult to make changes).

An ERD also helps you to communicate a proposed design to a non-technical client in a way that they can understand, which can help them recognize where their requirements may have been poorly communicated or misunderstood and, again, tweaks can be made before the database has actually been created, so we use <https://dbdiagram.io/> to create ERD.

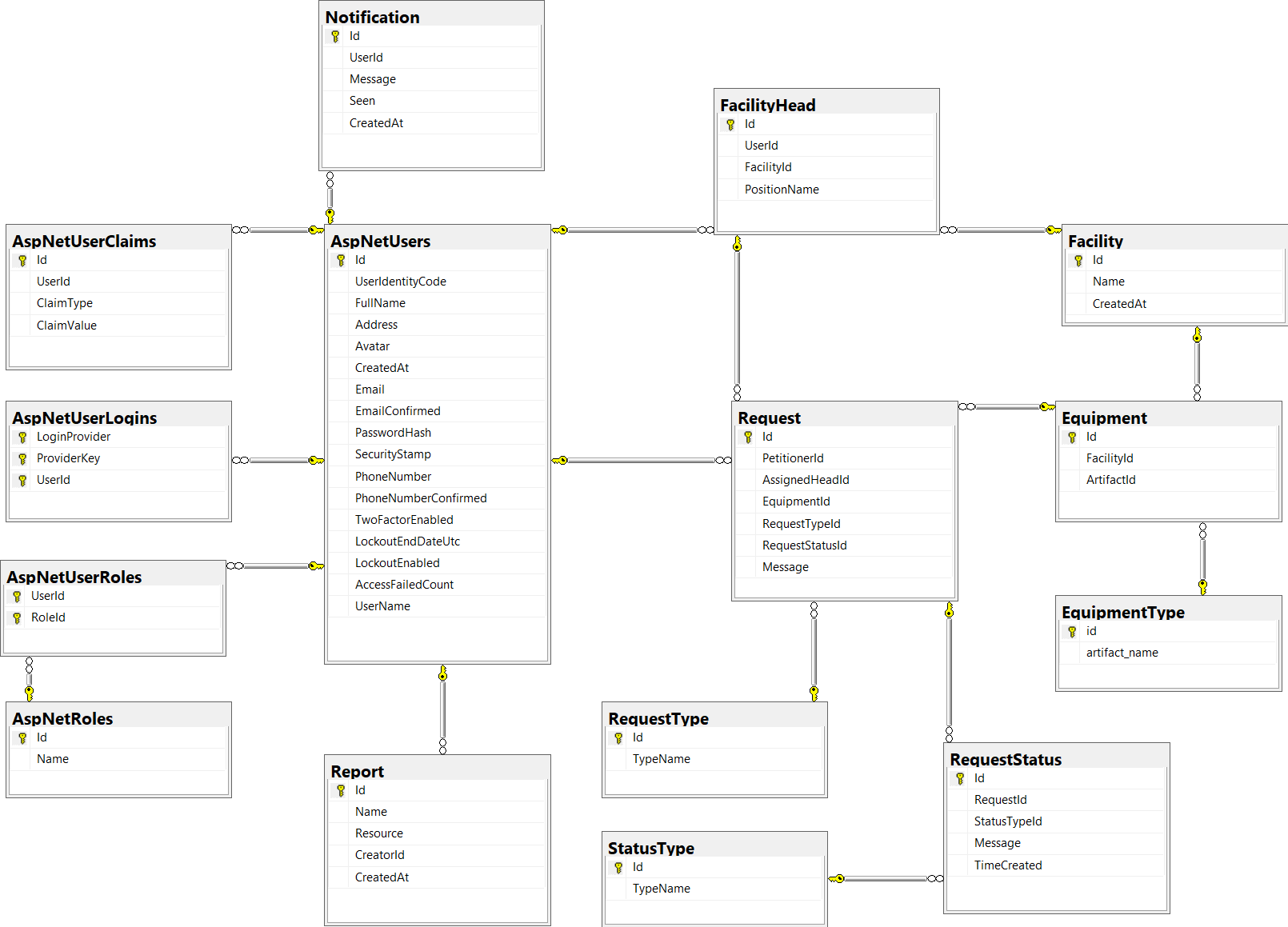


Figure 17. Entity Relationship Diagram of Online Help Desk system

## Database structure

The database table properties:

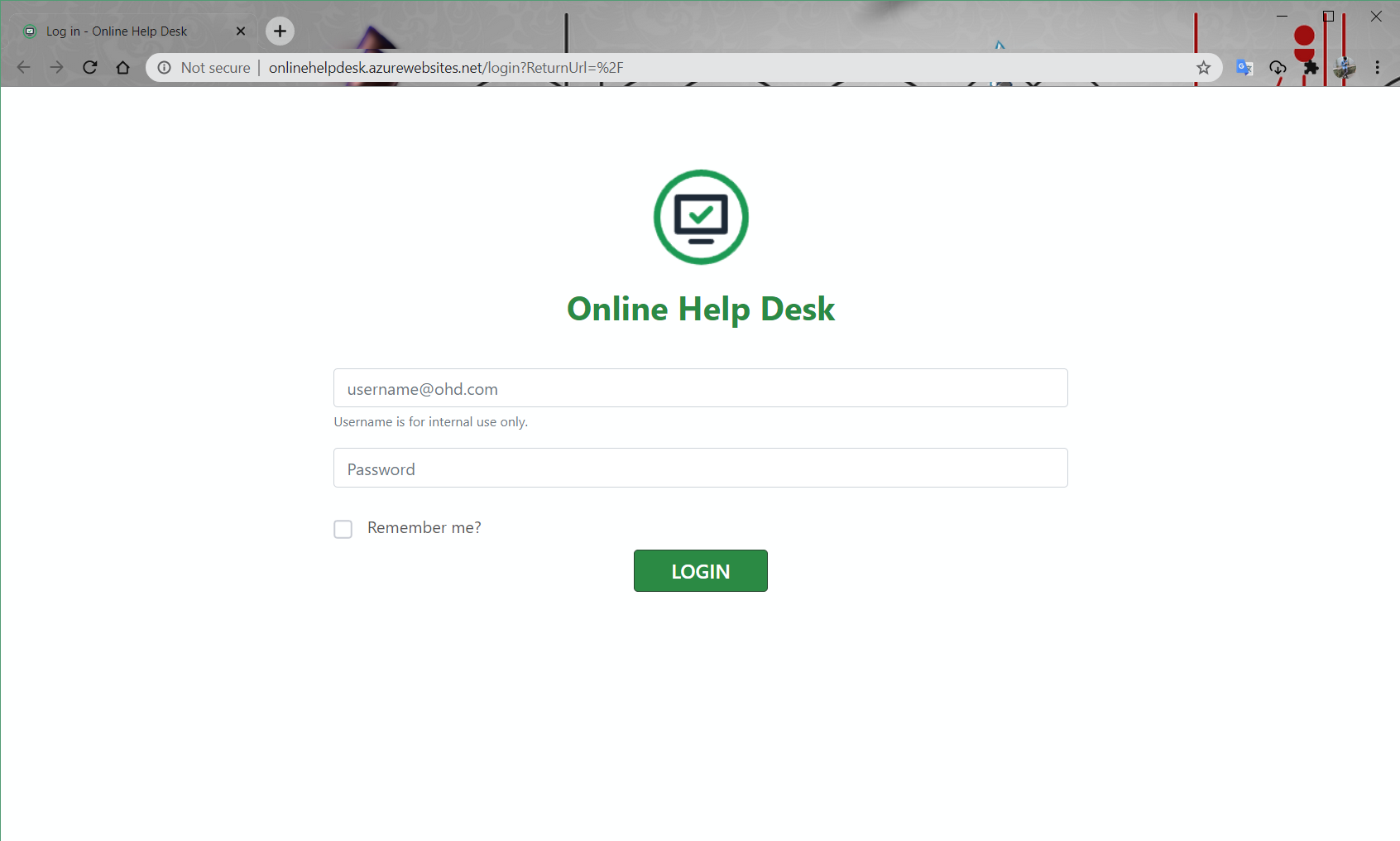
The system database is normalized and designed with the needed table that store the need information inside columns. Also, the need for developing the system capabilities and functionalities has been taken into account.

The system database tables created with the following structure:

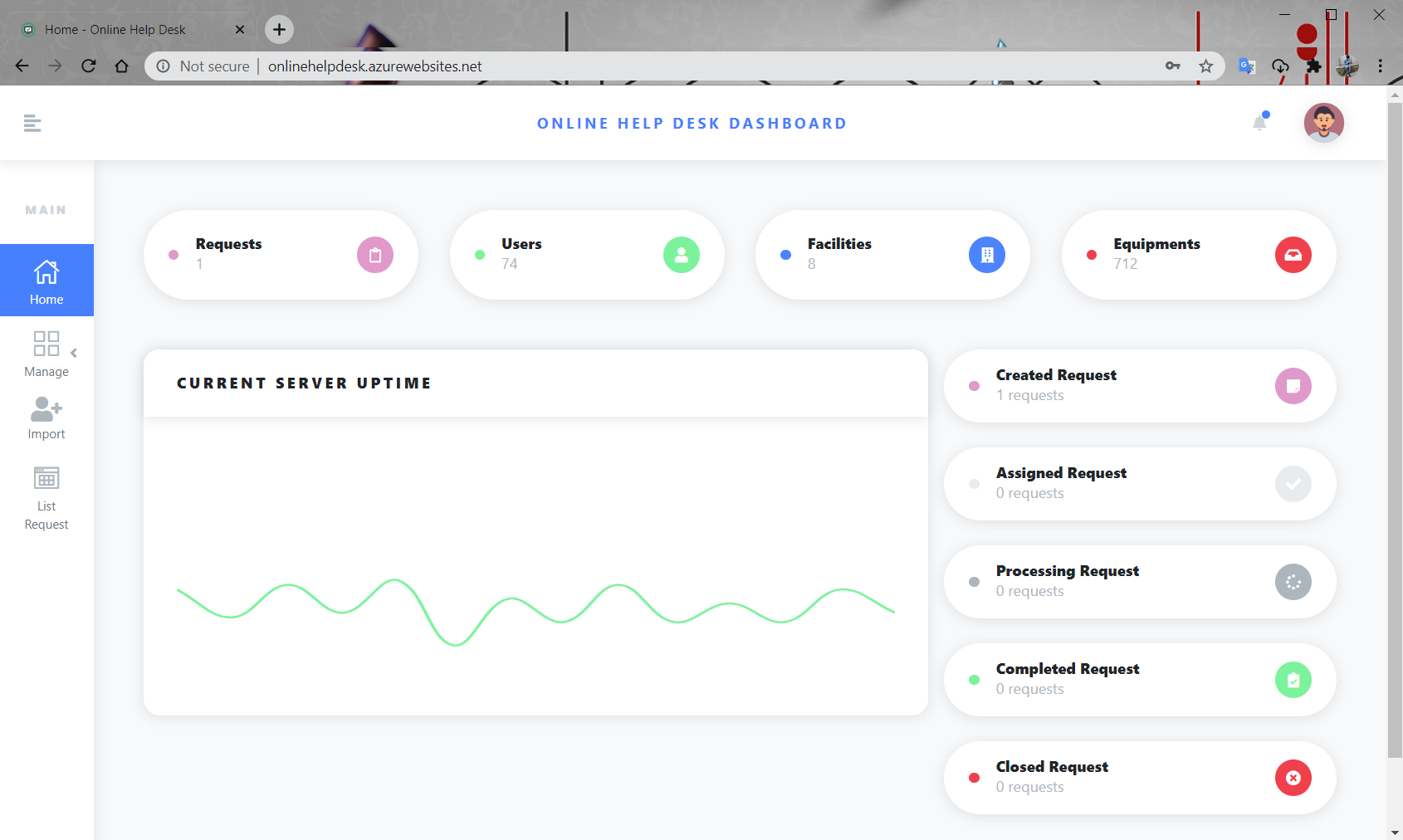
* Each table column is assigned with suitable data type
* Each table column is assigned with suitable data auto increment primary key
* For each table a foreign key is assigned when necessary plus its constraint type on Delete and on Update
* For each table an index and its type is assigned when necessary
* For each column a default value is set when necessary

## Screen shot

### Login page



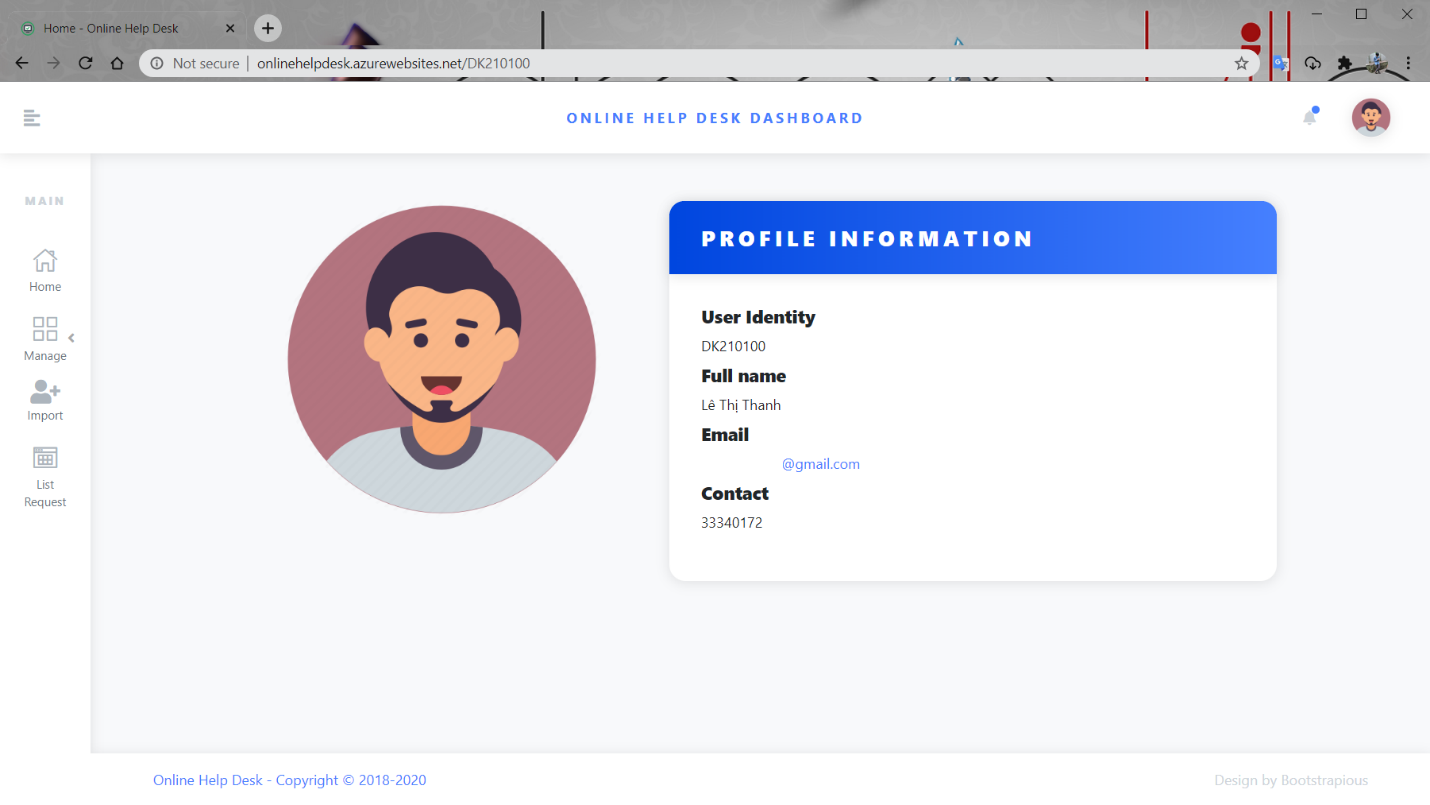
### Home page



### View users in role Page



### User profile page



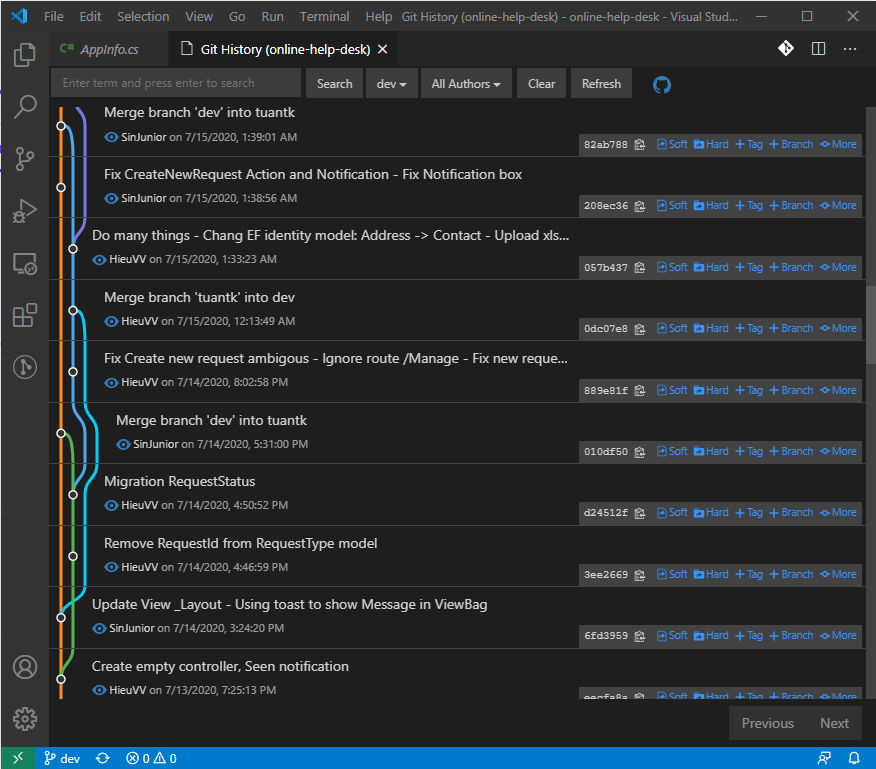
# Source code management

We use git at github.com to manage source code. Project’s repository is at: <https://github.com/voxvanhieu/online-help-desk>.

We have 5 branches at all, include:

* ***master*** for the cleanest code.
* ***dev*** for the working code and common branch for others team member.
* ***3 others member branches***

This is our git history graph



# Testing

## Purpose of Testing

The goal of testing was to demonstrate that the program under control contains bugs. Testing must not be confused with debugging, which is the process of detecting and reducing the number of existing errors. Testing can never prove that a code is error-, but rather verify that errors exist. Therefore, need to consider the fact that the error might come from the test itself, while the tested code might be correct. Have to know what the results of the test will show before it has actually been performed. The one who is responsible for doing the testing has to be able to define what the outcome should be, if not, this will lead to bugs in either the program or the test or in both the program and the test. The good thing about system testing is that it can be carried out without any prior knowledge about the program design and can thus be performed by “outsiders”. To maintain the quality of a system, it is definitely needed to configure a system testing. Detecting and fixing the errors in a system is known as one of the main objectives behind testing of a system in a development cycle. Here is described a set of test cases; some of them have been shown below:

* For checking the conjunction in the system that is developed, couples of users are connected to the Application simultaneously through the Internet browser. Different parts of system, project management and user management have been monitored simultaneously on these client machines. To conclude, the testing showed that system components could be monitored conjunctionally from multiple clients through Internet browser.
* It has also been noticed that data regarding system modifications are easily available with for all users. This affirms that the data recording feature of the developed system is working correctly.
* The connection has been tested from user application to commit code or make changes on code and later on to be seen on web browser. Commits and changes are nicely illustrated with differences made on the code on the web browser. These changes are seen for all users.
* Creating a new project and its administration, description and group. To add new project from admin panel to system, then we had to see, if the projects have made our repository. Where later on all updated files and commits have been added.
* Definitely, user management has been tested, while adding users, requests with different rights – belong to different groups.

## Non-functional Testing

Each one of the requirements need to have a way to evaluating if they have been successfully done or not. In this section, we will put the different tests that will be conducted to see if the previous requirements have been implemented successfully.

### Reliability testing

Reliability testing is used to find out the errors, bugs or failures before the deployments of the final product.

Unluckily, software reliability cannot be directly measured as of now. Other factors should be measured instead such as development process, faults and failures found.

We made ‘x’ number of tests (usually 7) and saw how many of them failed in some sort. When a requirement succeeded, we proceeded with the development.

### Usability testing

Usability is tested by testers to see how easy it is to use the system. It is a black box testing (internal structure/ design/ implementation of the item being tested is not known to the tester) which will also reveal how comfortable with our system they are. It will test how easy the website is to use and how easy is to learn to use.

The tests that we will perform are:

Take some testers and give them some tasks to do such as make an appointment, create an account or see the appointments made by them. It could be also done for doctors but with different things to do. While they do it, we time them and see how much time they needed to do the action. If they accomplished it in less than the amount specified in a requirement, than the test was successful.

At the end of the testing, each tester will give his/her feedback as to what things they found really nice and what things they found badly developed, hard to spot or just things to improve on in general Portability testing

We will see how the website functions on different devices. We used it on different browsers (Chrome, Firefox, Opera) and saw website’s performance on all of them.

### Performance testing

Performance testing tests how fast a system responds in different environments. It also measures the system workload.

Since we are limited in budget, we will test each functionality several times and compute the average time for each one of them. The average time should be less than 5 seconds.

And for concurrent users, we will just connect several users to a website at the same time and see how it performs. If all will be fine, the requirements will be satisfied.

### Security testing

Security testing checks whenever a software/website is secure or not by checking is it is vulnerable to attacks, of anyone can access a database or if anyone can login without authorization.

Several test cases exist to check some of the security of the website that we performed before deployment phase:

* Access bookmarked page without logging into the system
* Check password strengths
* Restricted pages can’t be accessed by restricted users

# Deployment

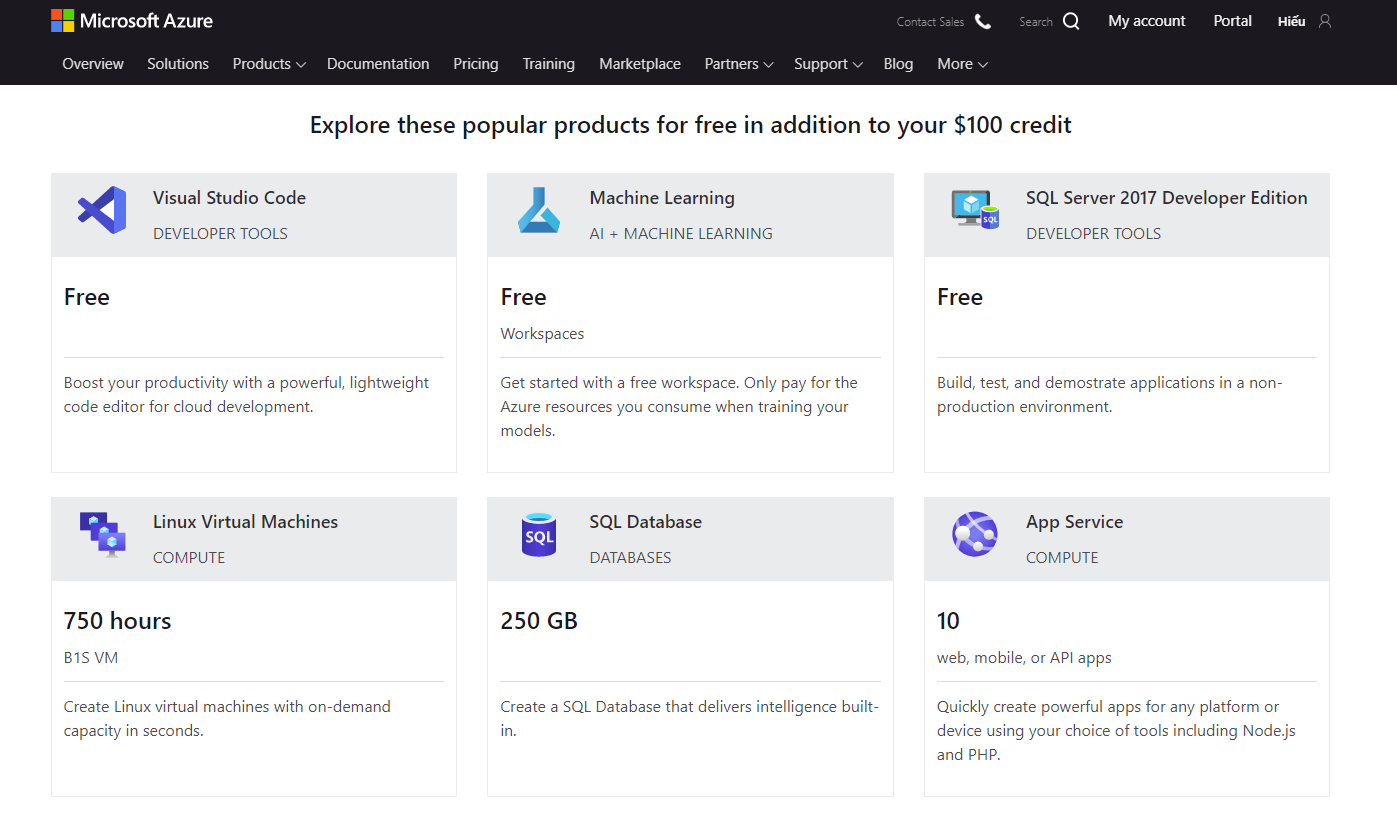
## Deployment Goals

As the scope of the project, we decide to publish the application to Azure Cloud services for demonstration purposes. Database and host are both on Azure. So, you can also access and test the project by this link: <http://onlinehelpdesk.azurewebsites.net/>

## Why on cloud?

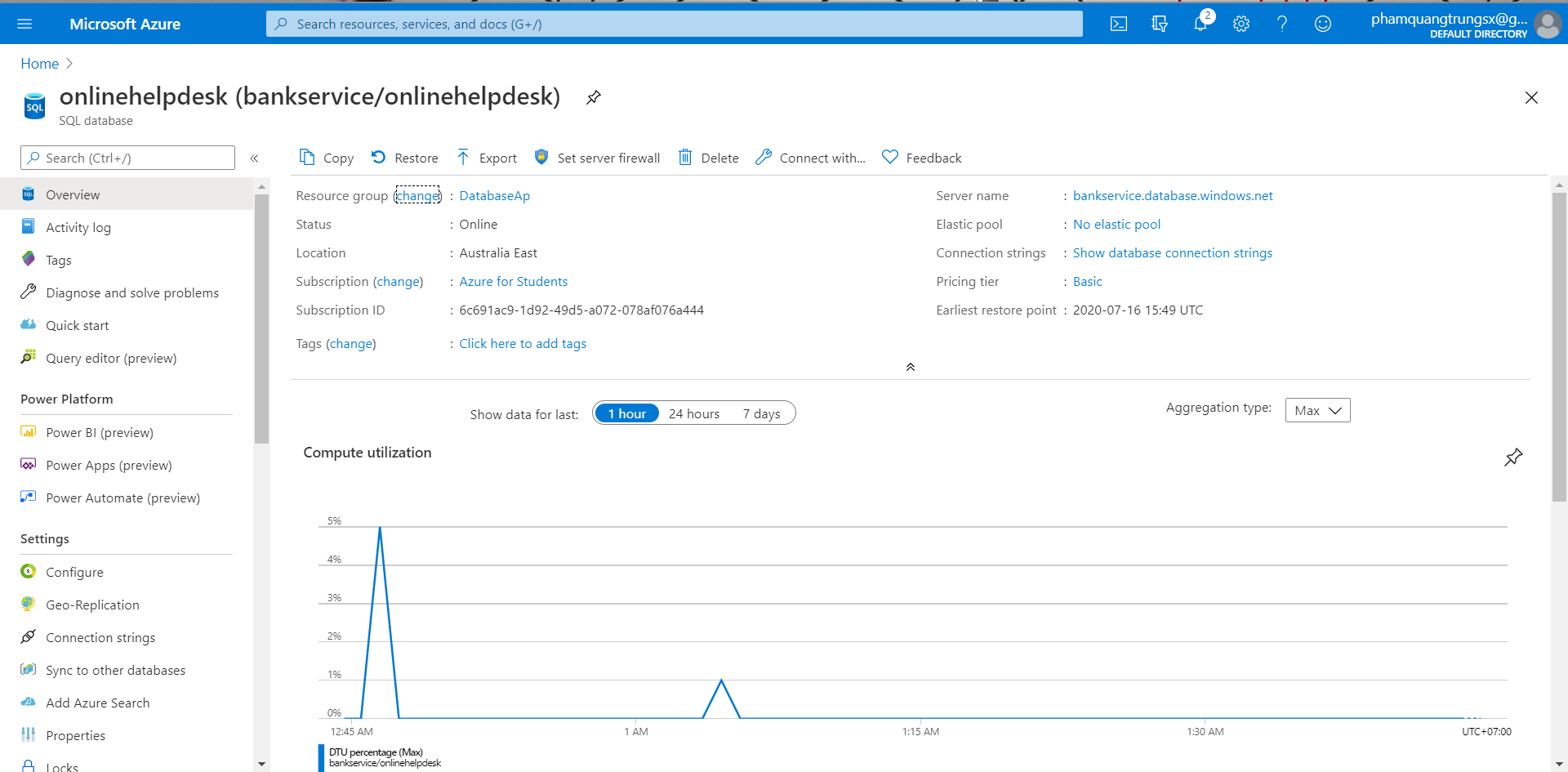
Nowadays, the rise of cloud computing and cloud service is the new trend. It brings back many benefits for individual users, enterprises, and students like us. With the free plan of Azure for Students, we can easily access and try out many incredible services of cloud computing. Create a database server, host a website, get a VPS. They are all easy.

As a part of Aptech learning curve, we decide to publish our ASP.NET project to Azure cloud.



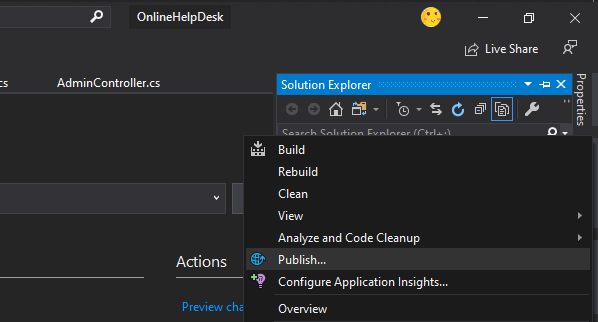
## Publishing steps:

### Export Profile of server and Database information on Azure

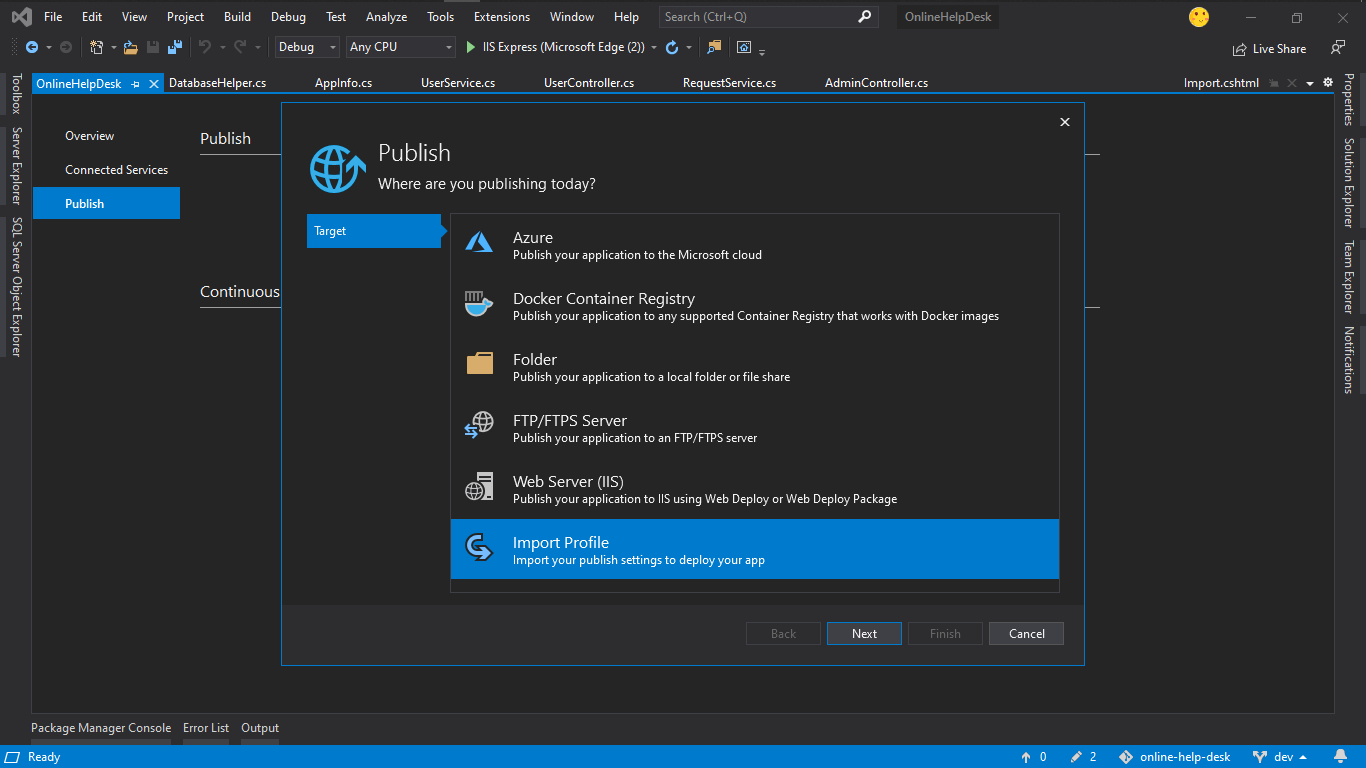


### Open project & Import publish settings

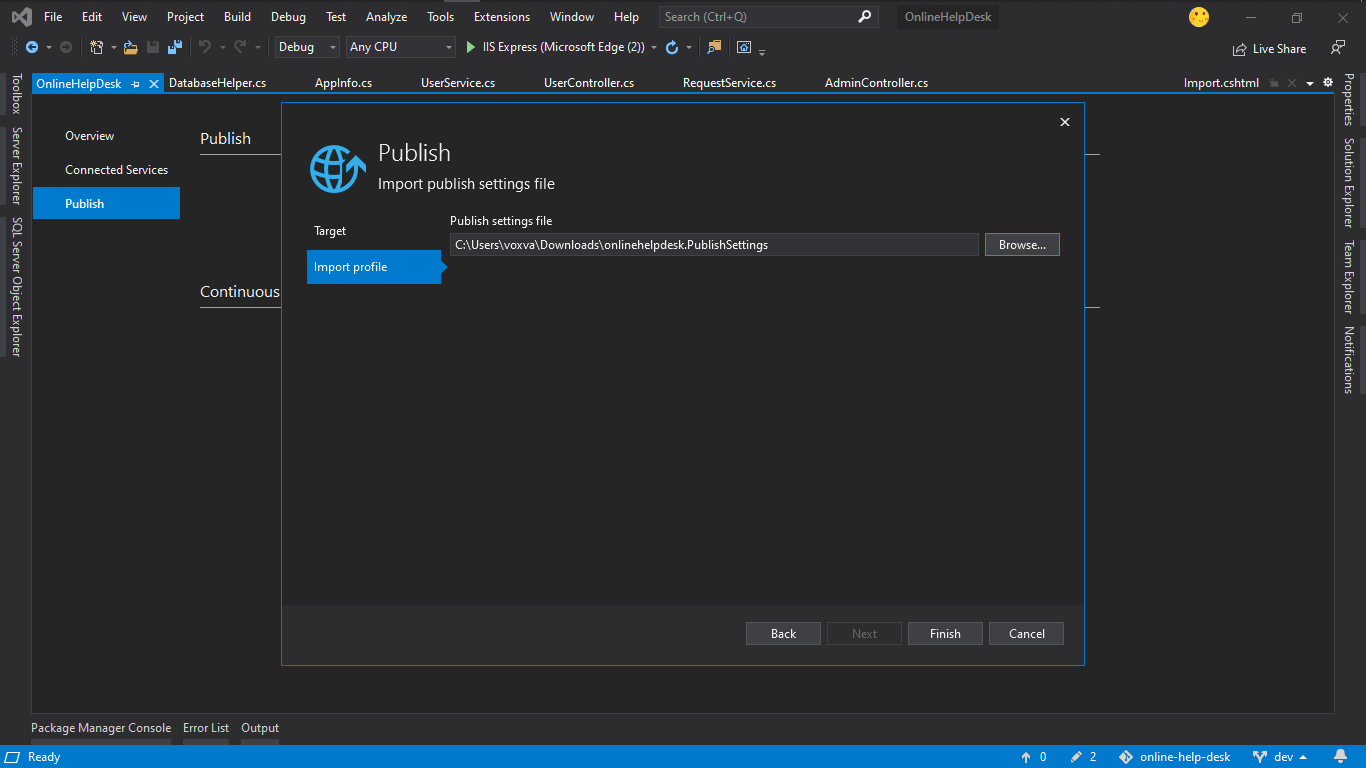
* Open project in Visual Studio 2019
* Right click and select ***Publish.***



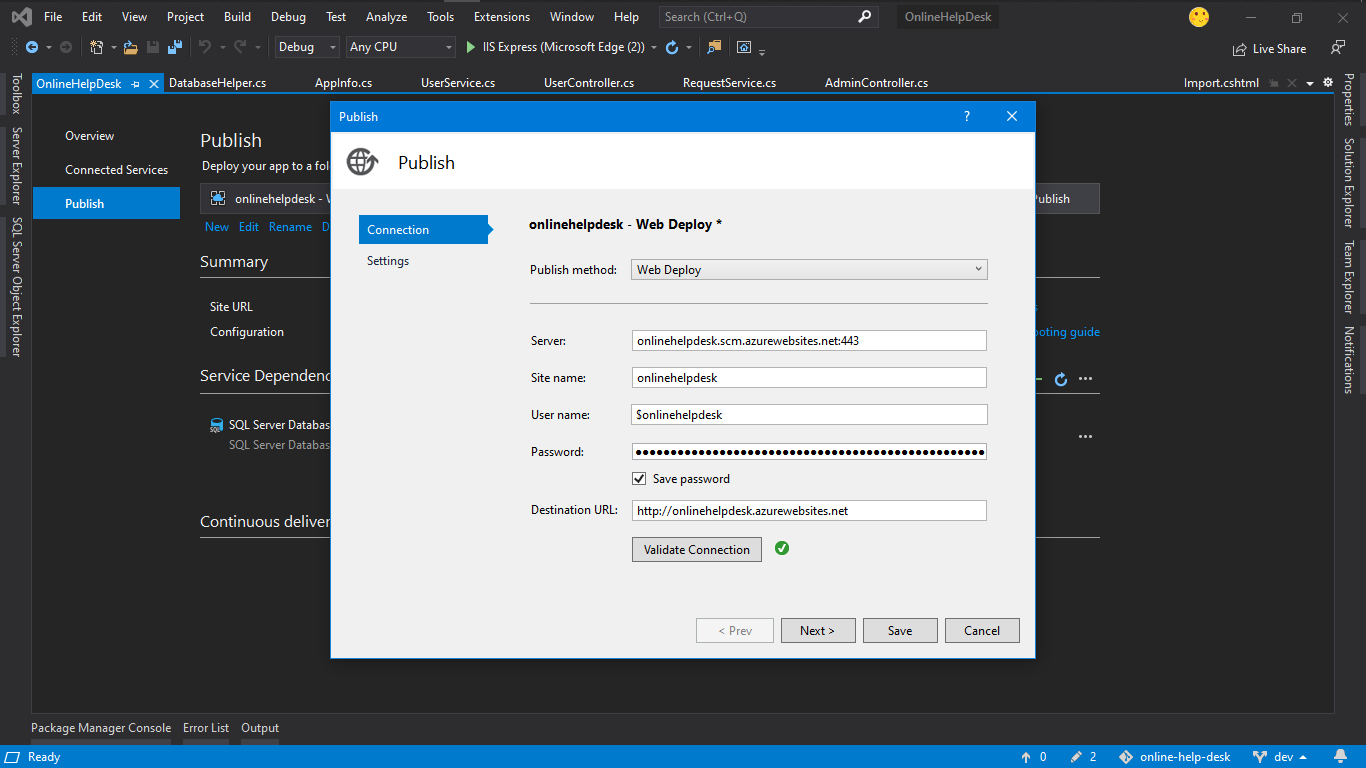
* Select Import Profile and click Next



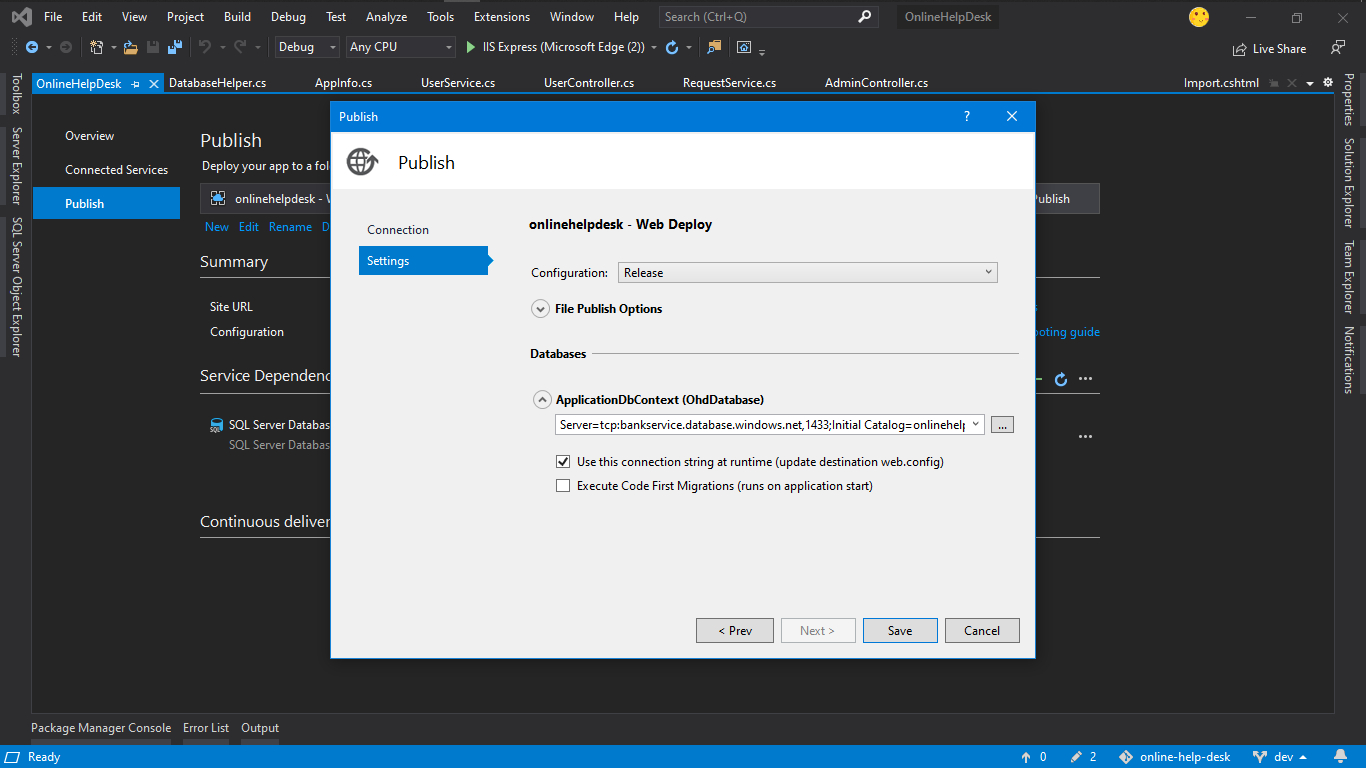
* Select ***\*.PublishSettings*** file and click ***finish***.



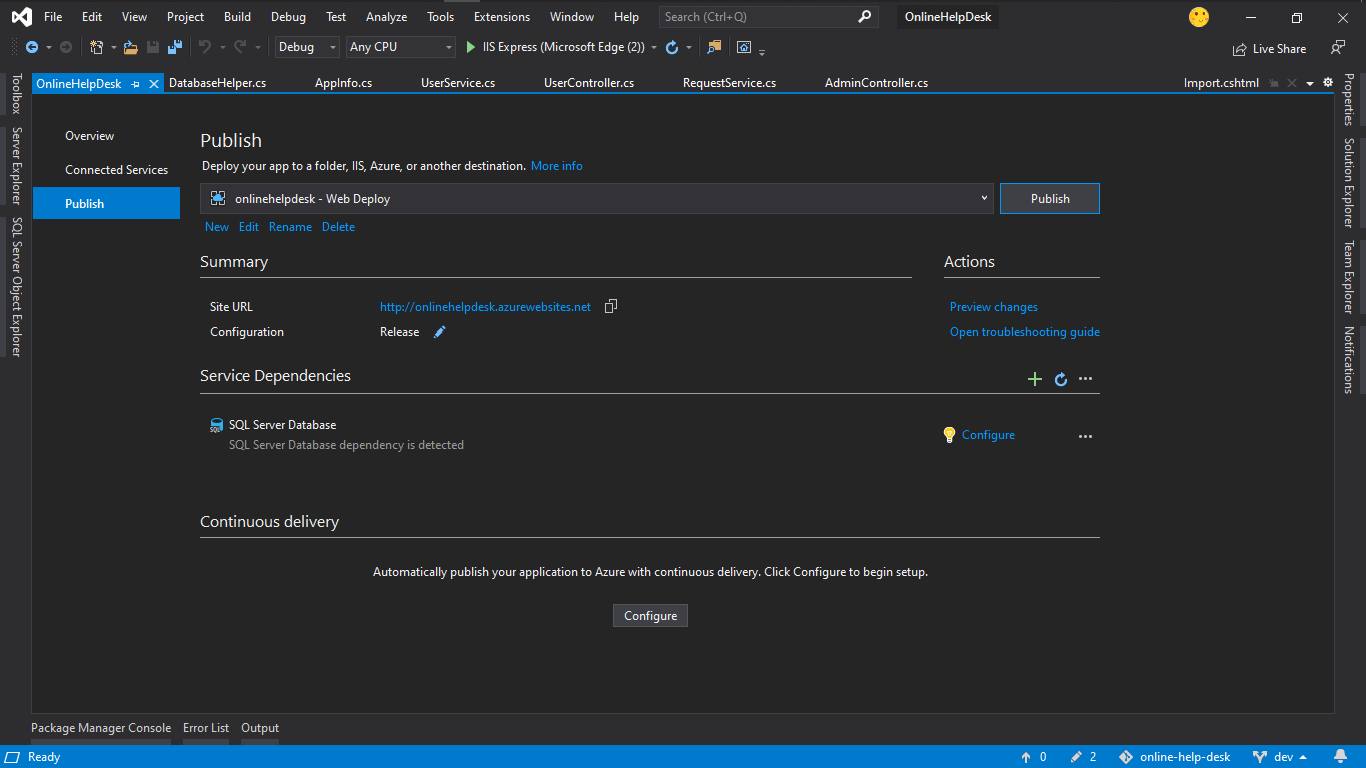
### Validates Connection:



### Paste the Connection string



### Publish page

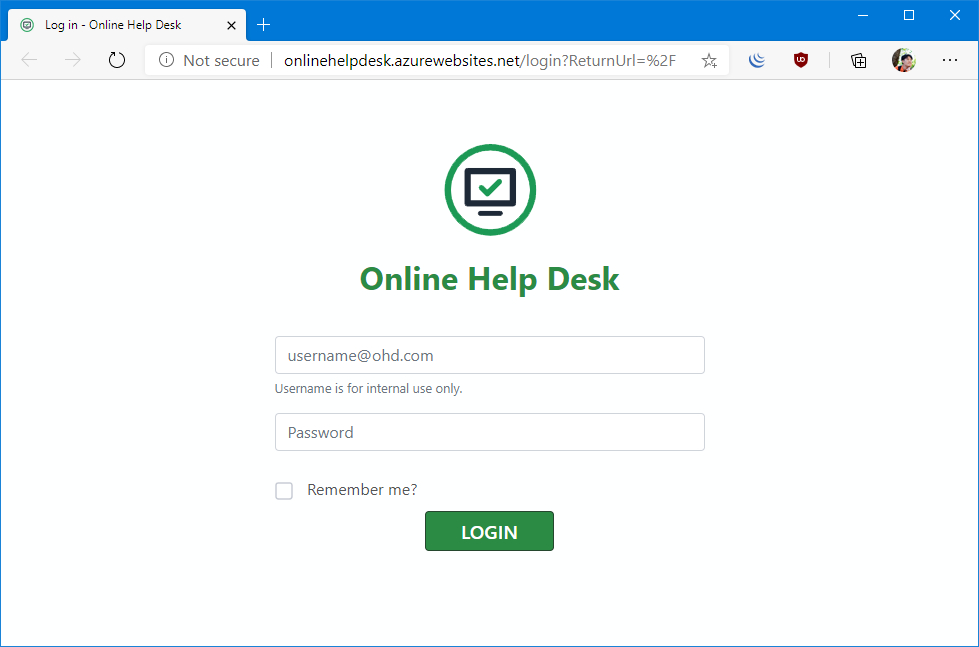


### Start publish



### Login page at first sight

After publish finished, the website can be accessible at: <http://onlinehelpdesk.azurewebsites.net/>



# Conclusions

The Online Help Desk (OHD) project is the small step to reduce the communication distance between the staff and the students.

As the growing use of computers and other electronic devices would mean the growing demand on rapid and quick technical support, this Online Help Desk System is carefully designed to fit with the rapid technical support. It not only helps reducing the time of recording and tracking inquires and problems traditionally, but also improves quality and accuracy of data produced by the system which can lead to more facilitation of decision making process in time.

OHD is designed to accommodate future upgrading and development without the need for building a new system to fit with the growing needs and demands of the system. Having this system hosted online means the ability of both technicians and administrator to track and respond to demands of students at any time beyond the boundaries and walls of college which add one more advantage to replacing the paper-based style.

Finally, the whole system has been tested to ensure that everything functions correctly before the system processes actual data and produces information that people will relay on.