Contents

[Project Phase 1 1](#_Toc436722974)

[Important Note 2](#_Toc436722975)

[Functional Requirements 2](#_Toc436722976)

[Non-functional Requirements 2](#_Toc436722977)

[Use Case Diagram 2](#_Toc436722978)

[Graphical User Interface 2](#_Toc436722979)

[Twitter Bootstrap 3](#_Toc436722980)

[SQL 3](#_Toc436722981)

[ERD 4](#_Toc436722982)

[LINQ Library 4](#_Toc436722983)

[GitHub 4](#_Toc436722984)

[Peer Review 4](#_Toc436722985)

|  |  |
| --- | --- |
|  |  |
| Due Date: | Friday December 18, 12PM |
| Percentage of module mark: | 30% |
| Submission Details: | Include all deliverables inside a zipped folder that is named with the members of your followed by \_projectPhase1. Your documentation must include:   * This document with each section completed. * Your MVC project containing your GUI and backend.   Marks will be deducted for additional documents submitted. |
| Late Penalty: | 20% deducted each day this assignment is late. |

## Project Phase 1

Make something that is commercially viable which interests your group and that would be seen as being useful to many others. Offer ideas that you do not mind sharing with your group. Your site must allow at least two types of users, it must use a database, and its commercial viability cannot rely on ad revenue.

The site should be designed so the business it conducts can be managed by one to two persons in their spare time while growing the business. The site should also be designed so the service you provide can be started up for basically no cost aside from modest hosting fees. You cannot rely on physical inventory or infrastructure to support this software. For example, the success of your software does not depend on a fleet of limousines, a team of bicycle specialists and bicycle inventory at a retail outlet, or a staff that is available by phone twenty four hours a day.

Do not make:

* A job board.
* A contract / contractee hub.
* A career portfolio host for job seekers or employers seeking employees.

## Important Note

Ensure your design models are compatible with each other and be careful not to over scope your project. Compatible models indicate team unity, clarity of direction, and well thought out solutions.

## Functional Requirements

Provide a current listing of functional requirements under essential, important, and nice-to-have categories. Please do this on one page - find a way to keep it organized and easy to read. 3 marks

Please show your functional requirements here:

|  |
| --- |
| Functional Features:  1. **Essential: (necessary for system to be useful)**    1. Logins that take consumers, business users, and administrators to their own area of website.    2. **Administrators** will be able to:       1. Manage consumer AND business user accounts       2. Create and add Smart-Device-Providers    3. **Consumers** will be able to:       1. Manage account details       2. Register available Smart-Device-Provider accounts       3. Register smart devices to the appropriate SDP account       4. View Device-Data-Report (analysis of user data)    4. **Business users** will be able to:       1. Manage account details       2. Subscribe to Anonymous-Device-Data 2. **Important: (can be added later)**    * 1. Consumers/Business users can view Device-Data-Reports in multiple formats 3. **Extras: (nice to have)**    1. Functional API’s from each provider (can alter devices states/status) |

## Non-functional Requirements

Provide a listing of non-functional requirements. 2 marks

Most of the time, this section may only include platform needs and colour / logo requirements. For larger systems and organizations, this section might include business policies, system workload restrictions, and security restrictions.

Please show your non-functional requirements here:

|  |
| --- |
| * Use SQL Server Database to manage data * Develop with VS using ASP.NET MVC, and AngularJS * Responsive Design (for all devices) * Use JS library to present data |

## Use Case Diagram

Please provide a current Use case diagram screenshot below: 3 marks

|  |
| --- |
| C:\Users\Matt Catellier\Dropbox\BCIT\MyCourses\InternalProject\use-case-diagram\use-case-v1.jpg |

## Graphical User Interface

Create an MVC project with a main layout page to store the tags that are shared by all pages. Add your GUI. The logic contained in the project will be incomplete but the GUI should represent the complete system for each type of user. Please also make it clear which page is viewable for each type of role. You may create separate pages for each type of user if this helps clarify the presentation (Redundancy is fine at this point if this helps you to clarify the different pages needed for the different types of users – we will look at how to streamline this later).

Do not waste time making graphics, fancy CSS, or javascript in this phase. Stay focused on page layout for link navigation, control alignment and position, and static realistic looking data content. (Do not use ‘lorem ipsum’ or data that has no meaning such as ‘asdf’ for the address. Do not create overly silly data such as user accounts for ‘Barrack Obama’ or ‘Mickey Mouse’ – this does not look professional.) Controls do not have to work but project stakeholders should be able to navigate through the page links to understand the site flow.

Create different versions of the site prototype for each type of user to show what each user type sees while navigating through the options that are available to them.

8 marks

Use proper alignment, sizing, and spacing. 5 marks

Your team must work with one project on GitHub. Please see the GitHub cheatsheet for a refresher on how to use this tool for sharing source code.

## Twitter Bootstrap

Use Twitter Bootstrap 3 for the majority of the formatting. Your implementation of Bootstrap can be very basic since we really have not covered it in class. Let’s start using it anyway so we can all become familiar with it. Implement a mobile responsive navbar which collapses in mobile view. 5 marks

Make your data look realistic. 1 mark

## SQL

In this document, include the SQL needed to build your ERD: 5 marks

Insert at least three rows of data for every table. 2 marks

The SQL must run from start to finish without error. 2 marks

**(Do not auto generate your SQL)**

To ensure your SQL is available to all team members, create an App\_Data folder for your ASP.NET project and place the file in there. This will allow you to keep your SQL with the rest of the project code on GitHub while you develop it.

Place your SQL Here:

|  |
| --- |
| GO  -- FK from Device  IF OBJECT\_ID('StoredData')  IS NOT NULL DROP TABLE StoredData;  -- FK from ProviderAccount, DeviceData  IF OBJECT\_ID('Device')  IS NOT NULL DROP TABLE Device;  IF OBJECT\_ID('DeviceData')  IS NOT NULL DROP TABLE DeviceData;  GO  -- FK from Account  IF OBJECT\_ID('ProviderAccount')  IS NOT NULL DROP TABLE ProviderAccount;  -- FK from Account  IF OBJECT\_ID('Details')  IS NOT NULL DROP TABLE Details;  IF OBJECT\_ID('Account')  IS NOT NULL DROP TABLE Account;  GO -- CREATE TABLES  CREATE TABLE Account (  accountID INT IDENTITY(1000, 1) PRIMARY KEY,  email VARCHAR(255) NOT NULL,  password VARCHAR(255) NOT NULL,  firstname VARCHAR(255) NOT NULL,  lastname VARCHAR(255) NOT NULL,  birthdate DATE NOT NULL,  type VARCHAR(255) NOT NULL,  CHECK (type IN ('Admin', 'Business', 'Consumer'))  )  CREATE TABLE Details (  accountID INT PRIMARY KEY,  familysize INT,  children INT,  adults INT,  rooms INT,  address VARCHAR(255),  FOREIGN KEY(accountID) REFERENCES Account(accountID)  )  CREATE TABLE ProviderAccount (  accountID INT PRIMARY KEY,  username VARCHAR(255) NOT NULL,  password VARCHAR(255) NOT NULL,  provider VARCHAR(255) NOT NULL, -- want to restrict somehow  FOREIGN KEY(accountID) REFERENCES Account(accountID)  )  CREATE TABLE DeviceData (  dataID INT PRIMARY KEY,  state VARCHAR(255) NOT NULL,  kwhour FLOAT(2) NOT NULL -- number with 2 decimals  )  CREATE TABLE Device (  deviceID INT PRIMARY KEY,  category VARCHAR(255),  accountID INT NOT NULL,  dataID INT NOT NULL,  FOREIGN KEY(accountID) REFERENCES Account(accountID),  FOREIGN KEY(dataID) REFERENCES DeviceData(dataID)  )  CREATE TABLE StoredData (  deviceID INT PRIMARY KEY,  state VARCHAR(255) NOT NULL,  datetime DATETIME DEFAULT(getdate()), -- stores both date and time  FOREIGN KEY(deviceID) REFERENCES Device(deviceID)  )  -- ADMIN  -- only needs to have a log in  GO -- Create account data  INSERT INTO Account VALUES('abc@123.com', 'admin', 'john', 'smith', '2015-11-14', 'Admin');  -- CONSUMER  -- will have login, provider log in, and device data  GO -- Create account data  INSERT INTO Account VALUES('john@hotmail.com', 'consumer', 'john', 'smith', '2015-11-14', 'Consumer');  INSERT INTO Details VALUES(1001, 4, 2, 2, 6, '1234 2nd Street');  INSERT INTO ProviderAccount VALUES(1001, 'john@hotmail.com', 'consumer', 'Nest');  GO -- create device data  INSERT INTO DeviceData VALUES(1, '78 Degrees F', 25.50);  INSERT INTO Device VALUES(1, 'Thermostat', 1001, 1);  INSERT INTO StoredData VALUES(1, '78 Degrees F', DEFAULT);  -- CONSUMER  -- will have login, provider log in, and device data  GO -- Create account data  INSERT INTO Account VALUES('bob@hotmail.com', 'consumer', 'bob', 'miller', '2015-11-14', 'Consumer');  INSERT INTO Details VALUES(1002, 4, 2, 2, 6, '1234 56th Street');  INSERT INTO ProviderAccount VALUES(1002, 'bob@hotmail.com', 'consumer', 'Samsung');  GO -- create device data  INSERT INTO DeviceData VALUES(2, 'On', 25.50);  INSERT INTO Device VALUES(2, 'Smart Bulb', 1002, 2);  INSERT INTO StoredData VALUES(2, 'On', DEFAULT);  -- CONSUMER  -- will have login, provider log in, and device data  GO -- Create account data  INSERT INTO Account VALUES('reggie@hotmail.com', 'consumer', 'reggie', 'bush', '2015-11-14', 'Consumer');  INSERT INTO Details VALUES(1003, 4, 2, 2, 6, '1234 New orleans Street');  INSERT INTO ProviderAccount VALUES(1003, 'reggie@hotmail.com', 'consumer', 'Nest');  GO -- create device data  INSERT INTO DeviceData VALUES(3, '54 Degrees F', 25.50);  INSERT INTO Device VALUES(3, 'Thermostat', 1003, 3);  INSERT INTO StoredData VALUES(3, '54 Degrees F', DEFAULT);  -- BUSINESS  -- will have login, and gather anonymous device data, no provider account  GO -- Create account data  INSERT INTO Account VALUES('david@hotmail.com', 'business', 'david', 'attenborough', '2015-11-14', 'Business');  INSERT INTO Details VALUES(1004, 4, 2, 2, 6, '1234 3nd Street');  GO -- View data  SELECT \* FROM Account;  SELECT \* FROM Details;  SELECT \* FROM ProviderAccount;  SELECT \* FROM DeviceData;  SELECT \* FROM Device;  SELECT \* FROM StoredData; |

## ERD

Show a screenshot of your ERD here. Please make it look professional in addition to ensuring it is relationally correct. 3 marks

|  |
| --- |
|  |

## LINQ Library

If you have time you can start writing the LINQ library. From past experience though, most teams would be better off to really polish their design and to ensure that all models are compatible with each other. No extra marks will be given for the creation of a LINQ library during this phase.

No marks

Teams are not permitted to share or distribute their code under any circumstances. However you may discuss code samples with other teams.

Marks will be deducted for code redundancy, non-standard code practices, and poor presentation.

## GitHub

Dec. 1 GitHub Requirement filled as outlined in the instructions. 3 marks

Significant continuous activity is logged in the history of the team repository. 7 marks

## Peer Review

5 marks

Everyone is to provide feedback for others in the team. Each survey is to be submitted anonymously. Once the results are in, you will only be able to view surveys for your own performance. Please keep your constructive and respectful - I am sure you will. If you wish to recommend something that can be improved that’s fine but also highlight things that your peers did really well. I will provide you with the following questionnaire on the final day.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Student Being Evaluated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_  Directions: Complete this form for each peer on your team. Rate each person using this scale:   |  |  |  |  |  | | --- | --- | --- | --- | --- | | Excellent  5 | Good  4 | Fair  3 | Poor  2 | Unsatisfactory  1 | | 10% of your project assignment grade will consist of a weighted average of evaluations from your team peers. You are to provide an anonymous rating for each of your team peers. Evaluation printouts will be provided on at the end of each phase when your goup work is submitted. Each student will be provided with anonymous feedback when they are evaluated. | | | | |  1. **Delivers appropriate levels of productivity for the team.**      1. **Shows professionalism towards other team members. Defends the team. Shows gratitude for their contributions, and does not criticize team members in front of others. Conducts constructive correction discussions privately with the person involved for their benefit.**      1. **Contributes to a healthy work environment by maintaining a professional attitude towards the project.**      1. **Expresses constructive differences of opinion when necessary but does not cause extended delays in production when differing from the rest of the team.**     **5. Advocates getting the job done properly and on time.**     1. **Shows voluntary initiative to support the team but also demonstrates good judgment to stay focused on their own project responsibilities.**      1. **Arrives to class at 9AM ready to help the team.**     **(Optional) Constructive Comments and Positive FeedBack** |