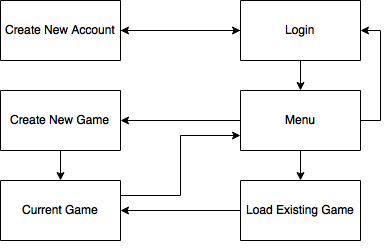
**Design**

**System Structure Diagram**

****

The first form will be the “Login” form in order for the player to be able to save and monitor their progress they must login to an existing account. If they are a new player and currently do not have an account there is a button to link them to the “Create New Account” form where they are able to enter in a new details such as a username and password so that they can logon in the future.

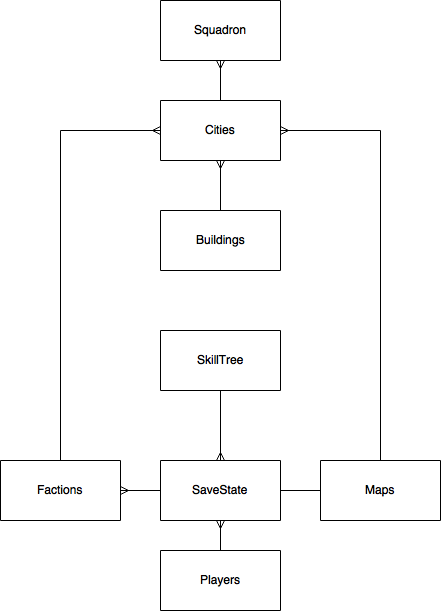
After logging on, the player is then redirected to the “Menu” form where there will be 2 buttons that the user can click, the first takes you to the “Create New Game” form and the second takes you to the “Load Existing Game” form.

The “Create New Game” form gives the player the ability to customize what they would like to play in the new game such as what faction that they would like to play as to what map they would like to play on. Additionally, they are able to choose what factions that they will be competing against and some information about their competitors. When they have finished choosing what they would like they can press the “Create Game” button which redirects them to the “Current Game” form.

Alternatively, the player can load an existing game within the “Load Existing Game” form. This form contains 3 buttons which load the game under different circumstances based on the save state that they contain. The form also includes some information about each save state in order to inform the player that they are selecting the right save state. After choosing which save state that they would like to load, they are redirected to the “Current Game” form.

Finally, the “Current Game” form is where most of the gameplay happens, the player is able to do a variety of things within this form such as choose what they would like to produce from a city by clicking on the tile that it is located on as well as the ability to see information like the amount of food, gold produced per turn as well as the total happiness of the city. When the player is happy with the choices that they have made they can then click next turn to see the outcome of their decisions, this process repeats until one of the three win conditions are met by any faction.

**Entity Relationship Diagram**

****

**Data Dictionary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: Players | | | Primary Key: AccountID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| AccountID | INTEGER | 2 |  | 1 | Unique Identifier. |
| Username | VARCHAR | 40 | Presence Check | BobSmith1348 | Allows the player to choose their own login name so it is easier to remember. |
| Password | VARCHAR | 25 | Presence Check | SecurePassword | Allows the player to choose their own password so it is easier to remember. |
| LastLogin | DATE | 10 |  | 22/07/2016 | Used when maximum numbers of accounts are in the database to delete accounts that have been inactive for more than 6 months. |
| Expert | BIT | 1 |  | 1 | Used to disable help messages if the player no longer requires them. |
| Wins | INTEGER | 2 |  | 5 | Tracks the win-loss rate of the player. |
| Losses | INTEGER | 2 |  | 0 | Tracks the win-loss rate of the player. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: Factions | | | Primary Key: FactionID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| FactionID | INTEGER | 2 |  | 10 | Unique Identifier. |
| Name | VARCHAR | 40 |  | The Steampunkers |  |
| Type | VARCHAR | 13 |  | Technological | Identifies what their most likely win condition will be as they will be more proficient in this area. |
| CapitalName | VARCHAR | 20 |  | Fallen London | The default name for the faction’s capital which the player can change. |
| Bonus | VARCHAR | 20 |  | Production | Used to make each faction a different playstyle. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: SaveState | | | Composite Primary Key: SaveID, AccountID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| SaveID | INTEGER | 1 |  | 2 | Limits the amount of saves that a player can have (3). |
| FileName | VARCHAR | 20 |  | OriginalFileName | Allows the player to implement a custom identifier to each save to inform them better. |
| AccountID | INTEGER | 2 |  | 19 | To link up save states to an account but also restrict those who do not have the correct AccountID from accessing other’s saves. |
| MapID | INTEGER | 1 |  | 3 | To link up a map to the save state’s information. |
| FactionID | INTEGER | 2 |  | 4 | Identifies what faction the player is in control of. |
| SkillID | INTEGER | 2 |  | 7 | Stores which skills that the player has unlocked during this game. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: Maps | | | Primary Key: MapID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| MapID | INTEGER | 2 |  | 5 | Unique Identifier. |
| Name | VARCHAR | 9 |  | Continent | Description of what the map is to use in a combo box for the player. |
| XCoordinate | INTEGER | 2 |  | 20 | Sets up the map grid. |
| YCoordinate | INTEGER | 2 |  | 14 | Sets up the map grid. |
| NumberOfTurns | INTEGER | 2 |  | 56 | Stores how many turns have passed since creation of the map (as there is a time limit). |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: Cities | | | Primary Key: CityID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| CityID | INTEGER | 1 |  | 9 | Unique Identifier. |
| Name | VARCHAR | 40 |  | Fallen London |  |
| XCoordinate | INTEGER | 2 |  | 5 | To establish where the city is located on the map. |
| YCoordinate | INTEGER | 2 |  | 6 | To establish where the city is located on the map. |
| TotalFood | FLOAT | 4 |  | 56.1 | Amount of food in surplus for citizens to eat. |
| FoodPerTurn | FLOAT | 4 |  | -1.4 | Can be negative. |
| TotalGold | FLOAT | 6 |  | 1000.6 | Amount of gold the city’s government has to spend on what the faction chooses. |
| GoldPerTurn | FLOAT | 5 |  | 20.2 | Can be negative. |
| TotalHappiness | INTEGER | 3 |  | 95 | How happy the people are under current leadership. |
| HappinessPerTurn | INTEGER | 2 |  | -1 | Can be negative. |
| FactionID | INTEGER | 2 |  | 7 | To identify current ownership of a city. |
| MapID | INTEGER | 2 |  | 2 | Identifies what map the city is founded in. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: Buildings | | | Primary Key: BuildingID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| BuildingID | INTEGER | 2 |  | 5 | Unique Identifier. |
| Name | VARCHAR | 25 |  | Armory |  |
| Description | VARCHAR | 40 |  | Reduced time and price for constructing armour and weapons. | Describes to the player the purpose of the building and why it should be constructed. |
| Type | VARCHAR | 13 |  | Militant | Informs the player what this building helps to achieve. |
| TurnsToConstruct | INTEGER | 2 |  | 14 | Stores the minimum time for a building to be constructed. |
| Constructed | BIT | 1 |  | 1 | Activator for when a city can finally benefit from the structure. |
| CityID | INTEGER | 2 |  | 6 | Identifies which city the building is in. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: Squadron | | | Primary Key: SquadID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| SquadID | INTEGER | 2 |  | 5 | Unique Identifier. |
| Active | BIT | 1 |  | 1 | Value to determine whether they are at base/dead (0) or active on the battlefield (1). |
| Objective | VARCHAR | 10 |  | Ambush | Command word for their objective for the player to choose what they want the squad to do (with a combo box). |
| XCoordinate | INTEGER | 2 |  | 5 | Stores the squad’s co-ordinates to actively track them on the map. |
| YCoordinate | INTEGER | 2 |  | 6 | Stores the squad’s co-ordinates to actively track them on the map. |
| CityID | INTEGER | 2 |  | 2 | Identifies which city the squad comes from. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table Name: SkillTree | | | Primary Key: SkillID | | |
| Field Name | Data Type | Length | Validation | Example Data | Comment |
| SkillID | INTEGER | 2 |  | 4 | Unique Identifier. |
| Name | VARCHAR | 25 |  | Strength In Numbers |  |
| Description | VARCHAR | 40 |  | Increases damage by 50% for each nearby squad. | Describes how the skill impacts gameplay to the player. |
| Type | VARCHAR | 13 |  | Militant | Separates skills based on win condition. |
| Obtained | BIT | 1 |  | 0 | Identifies if a skill has yet been unlocked. |
| ResearchTime | INTEGER | 2 |  | 6 | Stores how many turns it takes to research a new skill. |
| Tier | INTEGER | 1 |  | 1 | Limits a player to access skills until they have obtained the previous tier. |

**Description of Algorithms**

DDL:

|  |
| --- |
| Players Table |
| ADOCommand1.CommandText:='CREATE TABLE Players(AccountID INTEGER,' +  'Username VARCHAR(40),Password VARCHAR(25),LastLogin DATE,' +  'Expert BIT,Wins INTEGER,Losses INTEGER,' +  'PRIMARY KEY(AccountID))';  ADOCommand1.Execute; |
| Factions Table |
| ADOCommand1.CommandText:='CREATE TABLE Factions(FactionID INTEGER,' +  'Name VARCHAR(40),Type VARCHAR(13),CapitalName VARCHAR(20),' +  ‘Bonus VARCHAR(20),' +  'PRIMARY KEY(FactionID))';  ADOCommand1.Execute; |
| SaveState Table |
| ADOCommand1.CommandText:='CREATE TABLE SaveState(SaveID INTEGER,' +  'AccountID INTEGER,MapID INTEGER,' +  'FactionID INTEGER,SkillID INTEGER,' +  'FOREIGN KEY(AccountID) REFERENCES Players(AccountID),' +  'FOREIGN KEY(MapID) REFERENCES Map(MapID),' +  'FOREIGN KEY(FactionID) REFERENCES Factions(FactionID),' +  'FOREIGN KEY(SkillID) REFERENCES SkillTree(SkillID),' +  'PRIMARY KEY(SaveID,AccountID))';  ADOCommand1.Execute; |
| Maps Table |
| ADOCommand1.CommandText:='CREATE TABLE Maps(MapID INTEGER,' +  'Name VARCHAR(9),XCoordinate INTEGER,YCoordinate INTEGER,' +  'NumberOfTurns INTEGER,' +  'PRIMARY KEY(MapID))';  ADOCommand1.Execute; |
| Cities Table |
| ADOCommand1.CommandText:='CREATE TABLE Cities(CityID INTEGER,' +  'Name VARCHAR(40),XCoordinate INTEGER,YCoordinate INTEGER,' +  'TotalFood FLOAT,FoodPerTurn FLOAT,' +  'TotalGold FLOAT,GoldPerTurn FLOAT,' +  'TotalHappiness INTEGER,HappinessPerTurn INTEGER,' +  'FactionID INTEGER,MapID INTEGER,' +  'FOREIGN KEY(FactionID) REFERENCES Factions(FactionID),' +  'FOREIGN KEY(MapID) REFERENCES Map(MapID),' +  'PRIMARY KEY(CityID))';  ADOCommand1.Execute; |
| Buildings Table |
| ADOCommand1.CommandText:='CREATE TABLE Buildings(BuildingID INTEGER,' +  'Name VARCHAR(25),Description VARCHAR(40), Type VARCHAR(13),' +  'TurnsToConstruct INTEGER,Constructed BIT,CityID INTEGER, ' +  'FOREIGN KEY(CityID) REFERENCES Cities(CityID),' +  'PRIMARY KEY(BuildingID))';  ADOCommand1.Execute; |
| Squadron Table |
| ADOCommand1.CommandText:='CREATE TABLE Squadron(SquadID INTEGER,' +  'Active BIT,Objective VARCHAR(10), XCoordinate INTEGER,' +  'YCoordinate INTEGER,CityID INTEGER, ' +  'FOREIGN KEY(CityID) REFERENCES Cities(CityID),' +  'PRIMARY KEY(SquadID))';  ADOCommand1.Execute; |
| SkillTree Table |
| ADOCommand1.CommandText:='CREATE TABLE SkillTree(SkillID INTEGER,' +  'Name VARCHAR(25),Description VARCHAR(40), Type VARCHAR(13),' +  'Obtained BIT,ResearchTime INTEGER,Tier INTEGER,' +  'PRIMARY KEY(SkillID))';  ADOCommand1.Execute; |

**User Interface Design**

**System Security and the Integrity of Data**

Within the project, certain precautions have been taken to ensure that player usernames and passwords are kept secure and accurate. The project includes a doubly entry check of the user’s initial account creation to make sure their password is correct. Additionally, visual help will be included to give helpful messages to the user so that they don’t understand why they cannot yet enter the project.

To ensure referential integrity, if the player would like to delete their account they must first delete all their save states. If they attempt to delete their account with any save states still present they will be notified of why they cannot proceed.

As for data security, encryption will be used for the player and save state databases as they will contain private information about the player and manipulation of current save data is prohibited as it could be used to cheat.