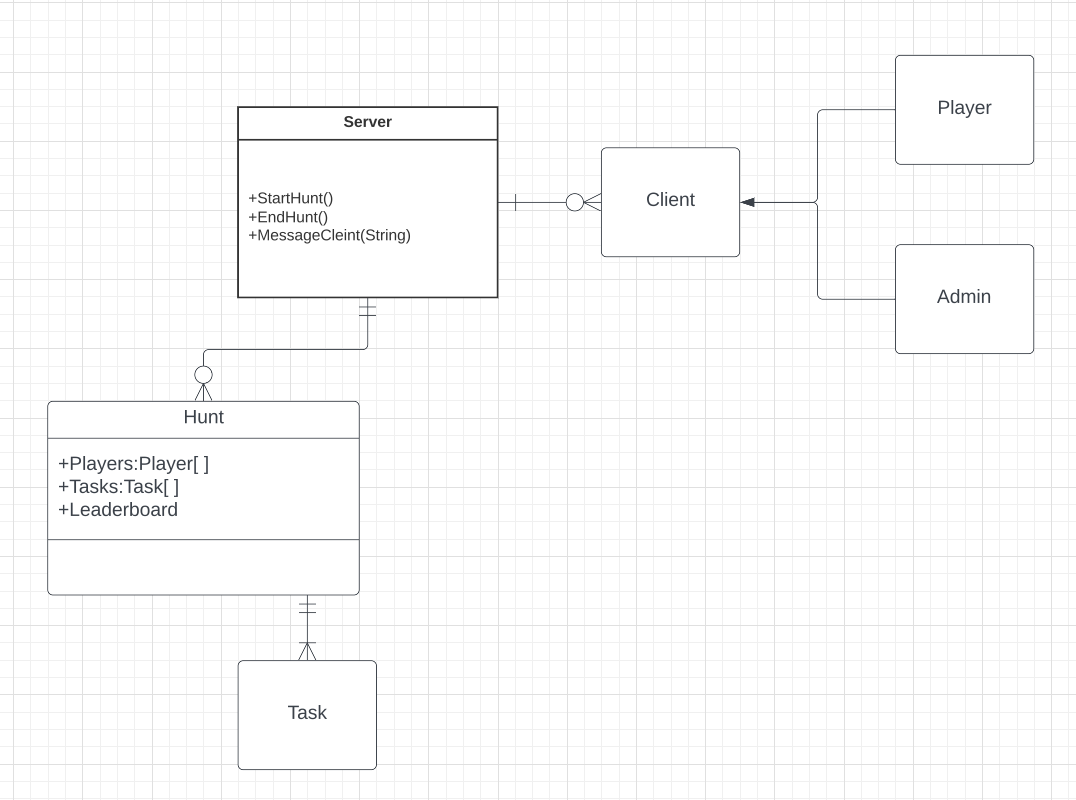
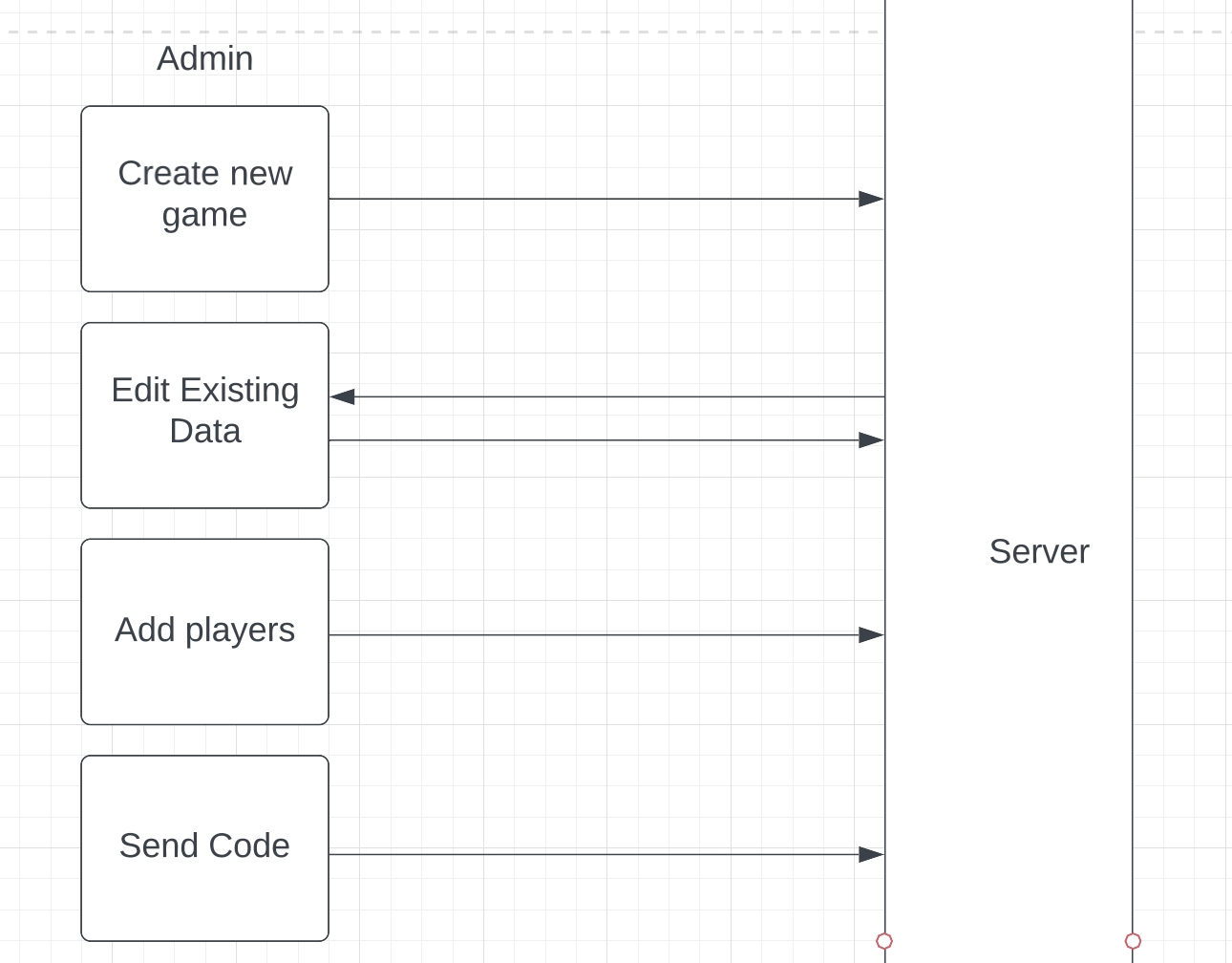
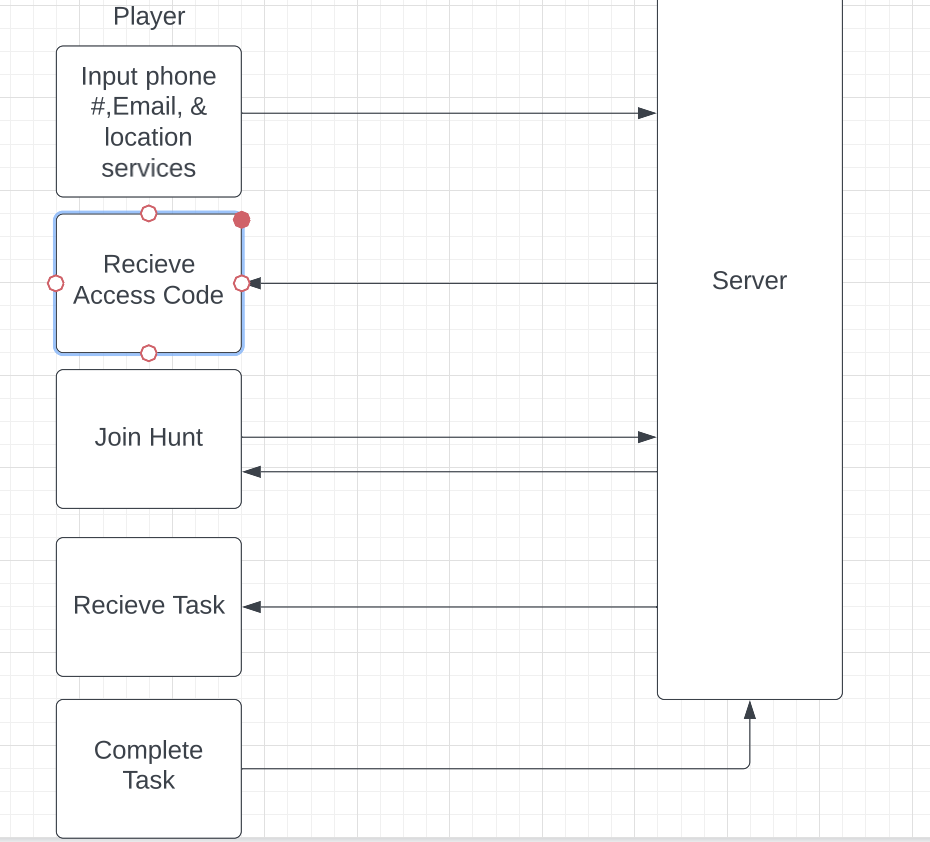
Super Awesome Scavenger Creation Club

Team: Stephen Maurer, Trevor Taylor, Reagan Mullins, Shaun Bennett, Aisara Imangaliyeva

1. Business Objectives –
   1. There are several business objectives for this product. First and foremost is to provide a more real-life development effort for students in this course. One which can be expanded on by subsequent semesters of students thus exposing them to having to work with an existing code base.
   2. Secondly, this product will provide ETSU with a more modern way to familiarize new students with the campus.
   3. Lastly, later renditions of this product have the potential to be a revenue generating solution for ETSU by allowing the university to white label it for other organizations to conduct treasure hunts outside of the campus.
2. Scope –
   1. The scope in iteration 1 is limited to the following:
      1. The list of tasks and player data must be persisted on a server.
      2. There is only one hunt enabled for this iteration and it is always “active”.
      3. Integration of QR Code scanning within the player interface is optional but desirable.
      4. Integration of a graphical map within the player interface is optional but desirable.
3. User roles — (name and briefly define/describe the user roles)
   1. Player: an individual (or team) who is playing / participating in the scavenger hunt
   2. Admin: person(s) responsible for updating the server-side information for the hunt and players. This person(s) will be responsible for sending any texts or emails to players that are not automated by the system.
   3. Player interface: the browser-based application the Player interacts with.
4. User functionalities – (list the product’s major user functionalities (functional requirements)
5. Server Architecture –
   1. Requirements -
      1. Creates Hunt object that stores all the different player and objective objects
      2. Objective objects are predefined tasks that are give to the Player
   2. Client Server Architecture -
   3. Admin Server Architecture -
6. BucHunt Database -
   1. User
      1. User Data(email, phone number, username, password, role(admin/user), group, history of locations,info etc)
      2. Email **and** phone number from player
         1. \*Must send users access code through text and email at same time with - “Thank you for Playing BucHunt: Your access code is <insert here> “)
      3. Access to location services on device being used on device
   2. Admin
      1. Account set up (username and password)
         1. \*whitelist certain emails to be admin

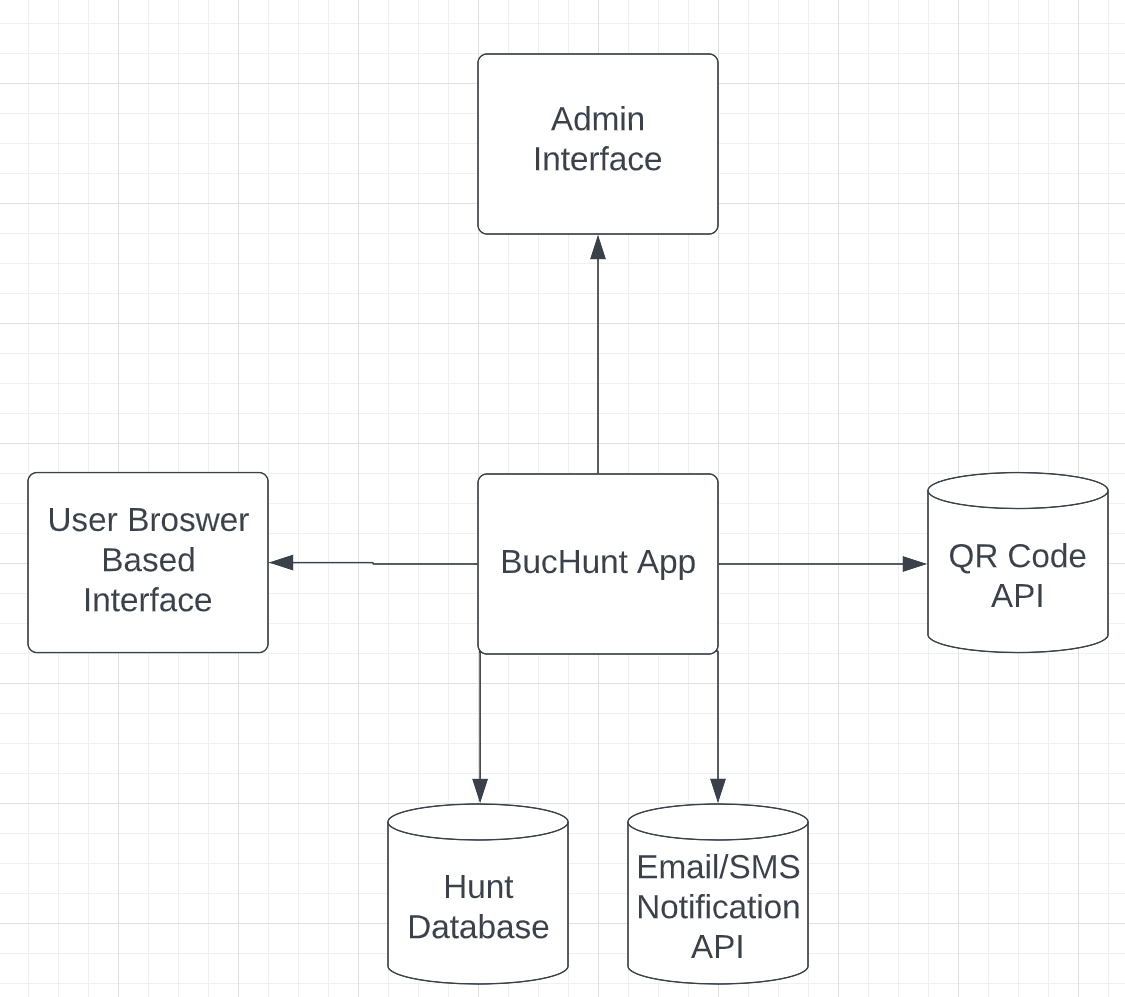




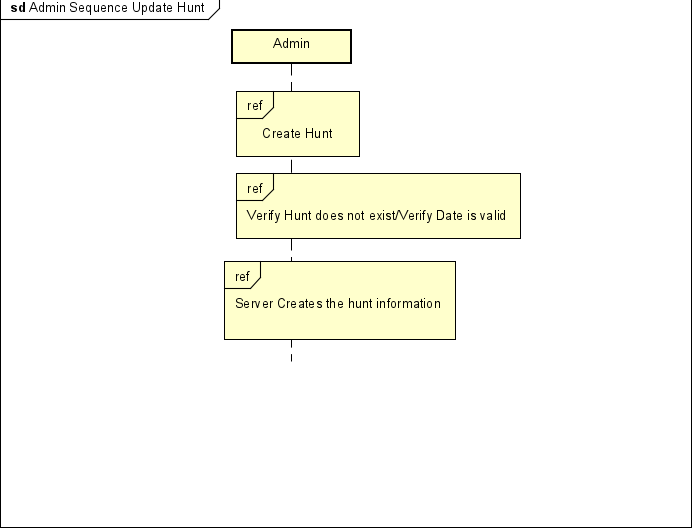


High Level Server Architecture

The BucHunt application will have several interactions with different APIs and have two distinct roles: users and admins. The major interactions with the system will be a QR code scanner api and an email/sms notification api to send messages to the list of users that will participate in the hunt. The user browser interface will be how the user will interact with the BucHunt app. The admin interface will be how the admin sends out messages to the users and add the objectives of the scavenger hunt.



Admin Create Hunt Sequence:



Admin Update Hunt Sequence:

