1. **Introduction**
   1. **Purpose of the system**

A Day in Bilkent is a shoot ‘em’ up style bullet hell game. Main purpose of the design is to make the game more enjoyable, and more challenging. The game is based on Bilkent. All the enemies, projectiles, companions are related to Bilkent. Player, which is a student in the game, has to overcome the quizzes, labs, and other assignments to reach the boss. After killing the boss, player can move to the next level. Our aims are improve player’s hand-eye coordination, fast decision making, making the player more challenging, and create an enjoyable game.

* 1. **Design goals**

**Game Performance:** We want our game as smooth as possible, because everything depends on movement in our game. So that game will be designed to conserve the frame rate. Hence user can play it easily. Moreover, we want to implement our game so that even outdated computers can run it easily.

**User-Friendly Interface:** The interface of the game will be simple. Player does not spend extra time to understand the game. Player can access the functions of the game easily so that it will create a user friendly environment. During the game run, player can see the health, score, and other components without breaking the concentration.

**Extendibility:** The game itself is open for additional extensions. Those extensions will improve the game and the gameplay experience. Changes in the companions, enemies, play modes, etc. will develop the players enthusiasm.

**Responsiveness:** Goal of the responsiveness is important for both us and players. We want to maintain the frame rate so that players can play the game without disturbance, because lag causes drop in concentration. Hence players do not get the joy of the game. Moreover, buttons will be implemented such a way that players do not get confused.

* + 1. **Trade-Offs**
       1. **Functionality vs. Understandability**

A Day in Bilkent depends on the abilities, characters, power ups, etc. so that functionality is enough to play the game. However, lots of functions force the players to learn all the functions of the game. We tried to make user friendly as much as we can. If the player knows the descriptions of the elements of the game, one can play better than others.

* + - 1. **Space vs. Speed**

Our game mostly depends on speed, so that we will use as much memory as we need. Game itself should be faster in order to maintain the competitiveness.

* + - 1. **Rapid Development vs. Functionality**

The game has lots of functionality to ease the gameplay. Thus, we have to code all of the functions, but functions take time so if we want to develop the game components it will take more time. However, varieties of the functions make the game more playable and fun. Hence players will enjoy the game.

* + - 1. **Programmability vs. Speed and Memory**

In game implementation, we focus on the speed, because speed is the keyword of our game. In order to increase the speed of the game, we have to use as much memory as we can, so that we do not push CPU to its limits, and everything will work as intended. However, to maintain the speed, we will code a lot. Hence, the program will be hard to change, because there will be lots of dependencies.

* + 1. **Criteria**
       1. **End User Criteria**
       2. **Maintenance Criteria**
       3. **Performance Criteria**
    2. **Definitions**