Sabotage
Secondary Appendix

Henry Huang

83420661

Team 11

Note that “…” refers to repeated fields and always appears after a previous field delimited by a comma.

HTTP Request Format

The messages contained in the HTTP requests are formatted in JSON strings.

De1-SoC POST

POST requests from the De1-SoC will be a subset of the following fields. The only mandatory field is the SessionID.

```
{
  "SessionID":"<Integer ID of the game session/lobby associated with the De1-SoC>",
  "Coord":
  {
    "lat":"<Floating point latitude>",
    "lon":"<Floating point longitude>"
  },
  "NumPlayers":"<Integer number of players in the lobby>",
  "Win":"<Integer user ID of player that found the De1-SoC>"
}
```

Mobile App POST

POST requests from the mobile app will be a subset of the following fields. The only mandatory field is the user ID. If SessionID is -1, the user is removed from their current session.

```
{
  "UserID":"<Integer ID of user>",
  "SessionID":"<The session ID of the lobby being joined>",
  "Sabotage":"<Integer number associated with an action>"
}
```

Sabotage actions should not be executed until after verifying the user is allowed to use that action.

Mobile App GET (Request Nearby Location Data)

These requests should be routed to /sessions. The user ID should be sent along with the GET request as a parameter. If it's a new user without an ID, it should have an ID of 0 and will get assigned a new ID. The user's GPS location must also be included as a parameter to get nearby sessions. The user ID (new or already existing) is returned with the rest of the HTTP response. These request will return the following fields.

```
{
 "UserID":"<Integer ID of user>",
 "Sessions":
 [
   {
     "SessionID": "<session ID>",
     "Coord": [lat, lon],
     "Metrics": {
       "NumPlayers": "<integer>",
       "TBD": <>
     }
   },
   ...
 ]
}
```

Get request format:

```
{
 "UserID": "<integer>"
 "PlayerLoc": ["<latitude>", "<longitude>"]
}
```

Mobile App GET (Request Current Lobby Status)

These requests should be routed to /lobby Poll whether the session a user is in is under the effects of a sabotage or if the requesting user won. It also returns the number of players in a specified session. UserID is a variable key determined by whichver users win.

```
{
 "SaboteurID": "<ID of whoever initiated the sabotage>"
 "Sabotage": "<integer>",
 "Duration": "<float (seconds)>",
 "NumPlayers": "<integer>",
 "SabTokens": <integer>,
 "Win": <bool>,
 "Leaderboard": [
   {<UserID>: <integer score>},
   ...
 ]
}
```

Get request format:

```
{
 "UserID": "<integer>",
 "SessionID": "<integer>"
}
```

MongoDB Format

Each collection is split into three clusters: `IDs`, `Users`, `Sessions`.

IDs Cluster

Essentially a table to see which IDs have been allotted to users already.

```
{
 {
  ID: <integer>
 },
 ...
}
```

Users Cluster

```
{
 {
  UserID: <integer>,
  NearbySessions: [
     {
      SessionID: <session ID>,
      Coord: [lat, lon]
     },
     {
      SessionID: <session ID>,
      Coord: [lat, lon]
     },
     ...
  ],
  CurrSession: {
   ID: <session ID>,
   Start: <float time when session began>,
   Timeout: <float time value when the CurrSession should be invalidated>,
   Win: <bool>
  },
    SabTokens: <integer number of sabotage tokens possessed>
    Timeout: <float time value when the user is counted as inactive and removed>
  },
  ...
}
```

Sessions Cluster

```
{
  {
    SessionID: <session ID>,
    Coord: [lat, lon],
    Metrics: {
      NumPlayers: <integer>,
      TBD: <>,
      ...
    },
    Sabotage: {
      SaboteurID: <User ID of whoever initiated the sabotage>,
      SabotageID: <Sabotage number>,
      Duration: <Time until sabotage expires>
    },
    LeaderBoard: [
      {<UserID>: <integer score>},
      ...
    ]
    Moved: <boolean>
  ...
}
```

The "Moved" field is not always present. It's purpose is to be present and "True" whenever the De1-SoC moves. Any "Moved" De1-SoC's must be checked manually and confirmed whether it is intentional or if it is theft. UserID is a variable key that is determined by whoever wins.