**Turn Waiting System App**

**Marcos Antoli | Carlos Fuentes | Cristian Garzon**

**marcos.antoli@upr.edu | carlos.fuentes1@upr.edu | cristian.garzon@upr.edu**

# **Introduction**

* Administrative transactions related to enrollment and coursework for students, known as regular user, at University of Puerto Rico at Mayagüez traditionally generate long queues with people standing in line. This is the specific problem we would like to tackle. Eliminating the necessity of waiting in line and having turns managed through the app. Also, the administrative personnel, known as administrative user, would control all turn cases that have been solved and are yet to be solved. The target users for this application are the students and administrative personnel. The application will be available as a web app and also as a mobile app. The mobile version will allow turn notifications to be sent to the mobile device natively as push while the web app will only display the information. Technologies used in this project are: java with the java play framework on the server side, html, css, javascript, bootstrap and angular js for the client side and lastly the relational database using postgresql.

# **Client App Description**

The regular user, the student, will be able to perform the following transactions:

* **User registration** where personal information and credentials must be provided, with confirmation via email.
* **User** **sign in** with the credentials that were provided upon registration.
* **Turn request** by indicating the following information: target department/faculty, purpose of turn request, and what type of transaction wants to be performed.
* **Current queue status** will be displayed specifying current estimated wait time.
* **Penalty fee payment** will permit to pay the fee before continuing to request another turn.

The administrative user will be able to perform the following transactions:

* **User registration** and **user sign in** will be done as well as a regular user.
* **Show queue** list to see how many regular users are waiting to get their turn and estimated time based on the queue.
* **Student Support** with a series of transactions which include:
* **Call User** - Alert the regular user that it is their turn.
* **Start Support** - to set the start time of the current regular user support time.
* **Add notes** - will allow the administrative user to add notes/comments to the current regular user turn.
* **End Support** - to set the end time of the current regular user support time.
* **Regular User No Show** - will allow the administrative user to mark the regular user as not present for those regular users that did not show up for their turn. The application will generate a penalty fee for the users that do not show up for their turn. When a regular user has a penalty fee, it must be paid in full before continuing to use the application.

# **Server Side Description**

Initially the server will store the data provided by the forms filled by users to be used at a later time. Upon user sign in, the user existence and credentials in the postgresql database will be verified. The server side will store all information and modifications, by the administrative user, related to the current turn on the database. The regular user will be notified when called by the administrative user. If the student does not show up when called, a penalty fee issue option must be available for the administrative user, and the regular user will have to pay the penalty before proceeding to use the system. The penalty fee payment method will not be stored in the database.

The tables that we are planing to use on the app are:

* **Users** for all the information about the students and administrative users.
* **Jobs** for saving all the information about the administratives jobs.
* **Roles** for specifying the difference between the different users.
* **Turns** for keeping a log of all the turns that have been ended.
* **Places** for saving the information about the places that you can ask for turn.
* **Penalties** for saving the information about the persons who don´t come to place when is his/her turn.
* **Payment** for saving the payment information relation with penalties.
* **Session** for saving when the session has been marked as *attending*, and when the session has been marked as *finished*.
* **Courses** for keeping track of all the courses you could request a turn to receive support.
* **Completed** for keeping track of all the courses a student has taken.
* **Curriculum** for keeping track of all the courses per curriculum available.

# **Division of Labor**

|  |  |  |  |
| --- | --- | --- | --- |
| Tasks | Carlos | Cristian | Marcos |
| User registration | **x** |  |  |
| User sign in |  |  | **x** |
| Turn Request |  | **x** |  |
| Current Queue Status |  |  | **x** |
| Penalty fee payment | **x** | **x** |  |
| Show Queue |  | **x** |  |
| Student Support - Call Student |  |  | **x** |
| Student Support - Start Support | **x** |  |  |
| Student Support - End Support | **x** |  |  |
| Student Support - Add Notes |  |  | **x** |
| Student Support - Student No Show |  | **x** |  |