Section 2.2 Development view

The development view describes our scheduling system from our programmer's perspective and the management of our software. It is illustrated by the component diagram which consists of components and links that show the dependencies between them.

<insert component diagram>

The main components of our system consists of the model, view and controller. Since these 3 components are not taken by the same framework, we have specified on the component diagram that the view is handled using React and that data manipulation and database queries are handled by Laravel. Most importantly, in our model, we have defined our subsystems to be account management, courses management, preferences management, schedule generation and database management. The different subsystems interact with each other through interfaces, with the open end of the link being a required interface and the lollipop being the provided interface. Also, the model is connected to a MySQL database. Where does security go in here.. ?

**Section 3.2 Subsystem Interfaces Specifications**

The subsystem interfaces consist of: ManageCourses, PreferenceSettings, Security, CourseAvailabilty, SettingImplementation, ScheduleGeneration. Each of the interface connect two subsystems together. In this section, each of these interfaces will be described and the function calls exchanged between its subsystems will be given along with the specific description of the parameters passed.

3.2.1

ManageCourses

The manage courses interface joins the provided interface of account management with the required interface of the subsystem courses management.

|  |  |
| --- | --- |
| **Classes involved** | Student, Course |
| **List of Methods** | 1. setTakenCourses(Course[]): void   **Implemented in Class**: Student  **Description**: Method to add courses to the list of taken courses.  **Input Parameter(s)**: an array of type Course  **Return Type**: void         2. getTakenCourses(): Course[]  **Implemented in Class**: Student  **Description**: Method to get the list of courses taken.  **Input Parameter(s)**: none  **Return Type**: an array of type Course         3. setNeededCourses(Course[]): void  **Implemented in Class**: Student  **Description**: Method to add courses to the list of needed courses.  **Input Parameter(s)**: an array of type Course  **Return Type**: void         4. getNeededCourses(): Course[]  **Implemented in Class**: Student  **Description**: Method to get the list of courses needed.  **Input Parameter(s)**: none  **Return Type**:  an array of type Course |

3.2.2

PreferencesSettings

The preference settings interface joins the provided interface of account management with the required interface of the preferences management.

|  |  |
| --- | --- |
| **Classes involved** | Student, Preferences |
| **List of Methods** | 1. setPreferences(Preferences): void   **Implemented in Class**: Student  **Description**: Method to set the preferences  **Input Parameter(s)**: object of type Preferences  **Return Type**: void   1. getPreferences() Preferences   **Implemented in Class**: Student  **Description**: Method to get the preferences  **Input Parameter(s)**: none  **Return Type**: An object of type Preferences |