**The Requirements Document:** IUStudent

**Abstract:** IUStudent is asmartphone application used, with a goal, to help students navigate, follow or organize their academic life according to their classes and/or events.

**Document Revision History:**

Rev0.1 March 20 ,2019 – initial version

**Glossary:**

**Introduction:** IUStudent is a primarily android mobile application, requiring an Android running phone in order to use it, version 5.0 or above (application would be acceptable for 86.2% of Android users). Since almost all IUS students carry their mobile phones with them, it would be more convenient to have all the student necessary information in one easily accessible place. IUS system isn't exactly known for its friendly user interface, especially when it comes to mobile phone use, it's clunky and often painful to use. Having a practical, functional and easily accessible user interface, specifically created for smartphone users would reduce the frustration and uneasiness dramatically. The main goal IUStudent is to help students organise their academic life, classes, events, homewards, quizzes and so on. Each student will have their own, unique profile, which there will be able to customize to suit there needs. Customisation can be done by following classes there are attending, which, if they enable to, will send them all the necessary information about the course, in form of a notification or it can be seen by looking at course profile, where they can look at its time table, assignments/homework, file or anything that professor wants to upload/share with students. Besides the courses student can follow the groups, there are run by an administrator or an appointed person, they will provide students with information regarding the group them/objective, for example, if student is interested in upcoming talks or events they can follow a group that will provide them with exact information, if form of a notification as soon as possible, or if they are interested in a student exchange program, they can follow the group for that, so they don’t miss out on any opportunities. All of this information that student is interested will automatically be added to their unique calendar, creating a visual picture of their everyday student life. Users can also look up things using the search option, possibilities are courses, classrooms, professor/staff and events that are in the database. Searching a course will provide the user with some basic information about it, place and time of lectures, the name of a professor who will teach the course and syllabus. Searching a professor will provide the user with some basic information about the professor, what course they are teaching, there email and credentials, and a time table with their lecture and office hours. Searching an event will provide the user with some basic information about the event, what is about, when it being held and where, duration and so on.

**System requirements specification:** This application will be build using Androids Java-based Android Studio, which is an IDE used for creating Android phone/tablet applications. The minimum version required to run IUStudent is 5.0, which is used by 86.2% of Android users. The application will be dependent on the internet, to log in, notifications and updates, but it can still be accessible offline, meaning that all the information, calendar and courses/group you are following will be there and accessible even without an internet connection. Each users profile will be protected by their own password, we can't vouch for 100% security, it depends on their password, and password could be changed by pressing the option “Forgot the password?” on the login screen.

**Functional Requirements**

**Viewer**

1. **Registration (must have):** Registration should only be possible by IUS students, requiring IUS student ID number, full name, email and password. Registration has to be approved by the Administrator or Professor, reason, to prevent a non-IUS student from registering as an IUS student.

**Student**

1. **Login (must have):** User needs to login to their created account, every user account is unique and information of it will be saved to server based on the user’s ID number.
2. **View profile (should have):** Ability for students to view and edit their profile, adding additional optional information, phone number, email. It can be seen by professors, and if set to the public also by other students.
3. **Search (must have):** Ability to search the database for courses, classrooms, events or professors, and provide the user with appropriate information related to item searched.
4. **Follow Course (must have):** Ability to follow the course student is interested or attending, by pressing the following checkbox, information about its schedule, time and place, will be automatically added to the user's calendar.
5. **Unfollow Course (must have):** Ability to unfollow the course student is no longer interested or attending, by pressing the following checkbox, information about its schedule, time and place, will be automatically removed from the user's calendar.
6. **Follow Group (should have):** Ability to follow the group student is interested in, by pressing the following checkbox, information about its schedule, time and place, will be automatically added to the user's calendar.
7. **Unfollow Group (should have):** Ability to unfollow the group student is no longer interested in, by pressing the following checkbox, information about its schedule, time and place, will be automatically removed from the user's calendar.
8. **Follow Event (could have):** Ability to follow an event student is interested in, by pressing the following checkbox, information about its schedule, time and place, will be automatically added to the user's calendar.
9. **Unfollow Event (could have):** Ability to unfollow an event student is no longer interested in, by pressing the following checkbox, information about its schedule, time and place, will be automatically removed from the user's calendar.
10. **Open Calendar (must have):** Calendar is personally weekdays plan for each student. It depends on the course, events or groups they are following or are interested in, and it will be updated/modified to any changes that user might make or professor.
11. **Set notifications (could have):** Ability to enable or disable notifications for each course, event or group the user is following.
12. **Sign Out (must have):** Users can leave their account whenever they want by pressing at the “Sign out”, button returning them to the login screen.

**Professor/Staff**

1. **Login (must have):** Professor needs to login to their created account; every professor's account is unique and information of it will be saved to server based on their ID number.
2. **Sign Out** **(must have):** Professor can leave their account whenever they want by pressing at the “Sign out”, button returning them to the login screen**.**
3. **Teach Course (could have):** Professor can choose what course there are going to teach, granting them with the ability to modify all the course information, add or remove links, change time and schedule, files and basic information.
4. **Add student to the course (would have):** Professorswould have the ability to add any student manually to the course.
5. **Remove Student from the course (would have**): Professors would have the ability to remove any student manually from the course.
6. **Add Course Information (could have****):** The ability for a professor teaching the course to add course information, add links, files or whatever the professor wants in the text, like or file form.
7. **Remove Course Information (could have):** The ability for a professor teaching the course to remove course information, remove links, files or whatever the professor wants.
8. **Modify Course Information (could have**): The ability for a professor teaching the course to modify any course information, modify links, files or whatever the professor wants.
9. **Cancellation of a Class (would have):** Ability professor have to cancel the class there are teaching, this option can only be approved by the administrator, and by cancelling it will remove all the information associated with the course.
10. **Add Quiz/Midterm/Final/Homework to Calendar (must have):** The ability for a professor teaching the course to add Quiz/Midterm/Final/Homework to the calendar of every student following the course. This will be followed by a pop-up notification.
11. **Remove Quiz/Midterm/Final/Homework from Calendar (must have):** The ability for a professor teaching the course to remove Quiz/Midterm/Final/Homework to the calendar of every student following the course. This will be followed by a popup notification.
12. **Modify Quiz/Midterm/Final/Homework on Calendar (could have):** The ability for a professor teaching the course to modify Quiz/Midterm/Final/Homework to the calendar of every student following the course. This will be followed by a pop-up notification.
13. **Share Events** **(would have):** The ability for a professor teaching the course to share the event they find it might interest students following the course. This will be followed by a pop-up notification.

**Administrator**

1. **Approve registration (must have):** Administrators ability to approve or dismiss the registration requests.
2. **Add Student (must have):** Administrators ability to add student manually to the database, used as a backup if there are some complications in the registration process.
3. **Remove Student (must have):** Administrators ability to remove a student manually from a database, used as a backup if there are some complications in the registration process.
4. **Modify Student Information (must have):** Administrators ability to modify student’s information manually to a database, used as a backup if there are some complications or errors.
5. **Add Professor/Staff (must have):** Administrators ability to add a professor manually to the database.
6. **Remove Professor/Staff (must have):** Administrators ability to remove a professor manually to the database, used as a backup if there are some complications or errors.
7. **Modify Professor/Staff Information (must have):** Administrator ability to modify professor’s information manually to the database, used as a backup if there are some complications or errors.
8. **Add Course (must have):** Administrators ability to add the course to the system/database.
9. **Remove Course (must have):** Administrators ability to remove the course from the system/database**.**
10. **Modify Course (must have**): Administrators ability to modify course information on the system/database.
11. **Add event (must have):** Administrators ability to add an event to the system/database.
12. **Remove event (must have):** Administrators ability to remove an event from the system/database.
13. **Modify event (must have):** Administrators ability to modify the information of any event on the system/database.
14. **Add category (would have):** Administrators ability to add a categoryto the system/database**.**
15. **Remove category (would have):** Administrators ability to remove a categoryfrom the system/database**.**
16. **Modify category (would have):** Administrators ability to modify categoryinformation no the system/database**.**
17. **Add group (would have):** Administrators ability to add a group to the system/database.
18. **Remove group (would have):** Administrators ability to remove a group from the system/database.
19. **Modify group (would have):** Administrators ability to modify a groups information on the system/database.

**Functional Requirements**

|  |  |
| --- | --- |
| **Primary Actor** | **Use Cases** |
| Viewer | 1. Register |
| Student | 1. Login 2. View profile 3. Search 4. Follow Course 5. Unfollow Course 6. Follow Group 7. Unfollow Group 8. Follow Event 9. Unfollow Event 10. Open Calendar 11. Set notifications 12. Sign Out   … |
| Professor/Staff | 1. Login 2. Sign Out 3. Teach Course 4. Add student to the course 5. Remove Student from the course 6. Add Course Information 7. Remove Course Information 8. Modify Course Information 9. Cancellation of a Class 10. Add Quiz/Midterm/Final/Homework to Calendar 11. Remove Quiz/Midterm/Final/Homework from Calendar 12. Modify Quiz/Midterm/Final/Homework on Calendar 13. Share Events   … |
| Administrator | 1. Approve registration 2. Add Student 3. Remove Student 4. Modify Student Information 5. Add Professor/Staff 6. Remove Professor/Staff 7. Modify Professor/Staff Information 8. Add Course 9. Remove Course 10. Modify Course 11. Add event 12. Remove event 13. Modify event 14. Add category 15. Remove category 16. Modify category 17. Add group 18. Remove group 19. Modify group   … |

**Use Cases**

**Register**

**ID:** FR1

**Priority:** High (Must have)

**Pre-condition:** Internet connection

**Post-condition:**  An account is created for the user

**Basic Flow:**

1. The user opens an App
2. The user click on ‘Don`t have an account? Sign up’ link below the Login button
3. The user enters the required information and clicks ‘Register’ button

**What can go wrong**

* If the ID already exists in a database, an error message will be displayed, warning the user and asking for a new ID or check with staff. If the ID does not exist, a confirmation message will appear, informing the user that the account has been created and approved.

**Login**

**ID:** FR2

**Priority:** High (Must have)

**Pre-condition:** Internet connection, User has an account

**Post-condition:**  The user is successfully logged into there account

**Basic Flow:**

1. The user opens an App
2. The user fills the necessary information in order to log in, ID and Password, and pressing the “Login” button to confirm the process of Logging in

**What can go wrong**

* The user could misspell there ID or password resulting in an error message telling the user that something went wrong. The user could forget the password, so there is an option “Forgot the Password?” at the bottom of the password text box if the user forgets his password he can change using his email

**View profile**

**ID:** FR3

**Priority:** Medium (Should have)

**Pre-condition:** An account

**Post-condition:**  Seeing their profile information

**Basic Flow:**

1. The user opens an App
2. User presses on the slide in the menu on the top left, the menu will slide from the left side to middle of the screen, the first top button sees “My profile”, pressing at that button will show the users profile

**What can go wrong**

* Lack of internet connection might prevent from accessing his/her profile

**Search**

**ID:** FR4

**Priority:** High (Must have)

**Pre-condition:** An account, Internet connection

**Post-condition:**  Receiving information requested

**Basic Flow:**

1. The user opens an App
2. The user presses the text field on top of the application interface
3. User types the thing he is searching
4. When a user types all of it in, the user should press enter to initialize the search

**What can go wrong**

* The user might misspell a word and it might impact the search results

**Follow Course (must have):**

**ID:** FR5

**Priority:** High (Must have)

**Pre-condition:** An account, Internet connection

**Post-condition:** Following checkbox is selected, users calendar is updated, the user receives the notifications about the course

**Basic Flow:**

1. The user opens an App
2. User presses on the slide in a menu on the top left, the menu will slide from the left side to middle of the screen, the second option from the top, button sees “Following”, pressing at that button will show all the courses on the right there checkboxes
3. The user will find the courses we are looking for and press the empty checkbox
4. Information that confirms that the user is now following the course

**What can go wrong**

* The course might not be set up properly

**Unfollow Course**

**ID:** FR6

**Priority:** High (must have)

**Pre-condition:** An account, Internet connection

**Post-condition:** Following checkbox is unselected, users calendar is updated, the user won't receive the notifications about the course

**Basic Flow:**

1. The user opens an App
2. User presses on the slide in a menu on the top left, the menu will slide from the left side to middle of the screen, the second option from the top, button sees “Following”, pressing at that button will show all the courses on the right there checkboxes
3. The user will find the courses we are looking for and press the activated checkbox
4. Information that confirms that the user is no longer following the course

**What can go wrong**

* The course might not be set up properly

**Follow Group**

**ID:** FR7

**Priority:** Low (should have)

**Pre-condition:** An account, Internet connection

**Post-condition:** Following checkbox is selected, users calendar is updated, the user will receive the notifications about the Group

**Basic Flow:**

1. The user opens an App
2. User presses on the slide in a menu on the top left, the menu will slide from the left side to middle of the screen, the second option from the top, button sees “Following”, pressing at that button will show all the groups on the right there checkboxes
3. The user will find the group they are looking for and press the activated checkbox
4. Information that confirms that the user is now following the group

**What can go wrong**

* The course might not be set up properly

**Non-Functional Requirements**

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Definition** | **More Details** |
| NFR1.: Hardware interface – Server Side | The ability for the server to hold text information. | Operating System: Linux  Disk Space: < 2GB |
| NFR2.: Hardware interface –  Client Side | A smartphone running Android 5.0 version or above. | Operating System: Android 5.0 or higher  Disk Space: < 100MB |
| NFR3.: Communication Interface | Internet connection would be necessary in order to communicate with a server. | Internet connection |
|  |  |  |