1. Project proposal
   1. Introduction

Addis Ababa University (AAU), established in 1950 is the oldest and largest higher learning and research institute in Ethiopia. Today, AAU has more that 33,000 undergraduate students and over 2,000 academic staff in its 14 campuses.1 Creating a conducive learning environment is a multifaceted challenging feat. Part of the challenge comes from the number of moving parts involved in making the University function in harmony — the academic staff, supporting staff, student body, administration, leadership and many more.

One of these parts, important in the teaching-learning process, is the communication between students and their instructors. Most commonly, instructors relay message to students through a class representative, an assigned student for each section. This representative has the responsibility of making sure every student in the section receives the announcement, the handout copy or any other information. Aside from the class representative, staff members use the noticeboard to post announcements.

In November of 2015, during our first year at the Department of Computer Science, we noticed the inefficiency of the communication. With this problem in mind, we built AAU Push. A website that gives teachers a simple way to post announcements and upload course textbooks, references and assignments for students. Students receive these information through the website and the Android application. A similar system, made by an American company, called Piazza ([piazza.com](http://piazza.com)) is used by the Software Engineering Department at Addis Ababa Institute of Technology.

The utility of AAU Push was evident over the following two years as the staff of Department of Computer Science used it to communicate to students. It relieved unnecessary pressure from the class representative and empowered instructors to be more interactive outside of class. Currently this system is to be placed to all departments in the College of Natural and Computational Sciences (CNCS) under the recommendation of the dean of CNCS.

Although this system is good and helpful, it is not without shortcomings. In this project we seek to find out problems in this current system and improve its usability, practicality and adoption.

1 <http://www.aau.edu.et/about/aau-at-glance/>, accessed December 27, 2018

* 1. Statement of the Problem and Justification

AAU Push (Push) currently requires both instructors and students to signup and login to use the website. There, instructors share information, announcements or files like assignments, references and PDF handouts.

The system is faces multiple limitations:

* Communication is one way: instructor to student
  + - * Students don't have a way to ask questions or submit assignments
* The system works over the internet. If file sizes exceed a certain limit, downloading the resources will be difficult. Large file sizes also incur higher costs for server space.
* Although the system has an Android app at some time, it is now out of date and useless.
* For iOS users, an application is not available.
* Several usability issues have been reported.

In addition to these limitations, there are many more features that could enhance the teaching-learning experience. Fixing these issues and developing the system further will help students be well informed and productive because AAU Push delivers, in one place, all information from teachers in the most organized manner. It will also help instructors be at ease and reassured because Push lets them reach all their students in the fastest, reliable and convenient way from anywhere at anytime.

* 1. Project Objective
     1. General Objective of the System

The general objective of this project is to create e-learning communication platform to aid the instructor-student relationship.

* + 1. Specific Objective of the System
* Document and analyze the current system
* List the shortcomings to the current system
* Gather list of features to add and errors to correct
* Develop features as needed
* Test and deploy the new system
  1. Scope of the Project

//We will be back to this after seeing the requirements and decide on what we will do and not do.

* 1. System Development Methodology

Since our system is driven by user interface requirements, we will be using Rapid Application Development as our software development approach. We will be putting less emphasis on planning and more emphasis on an adaptive process — prototyping, testing and documenting when approved.

* + 1. Investigation (Fact-Finding) Methods

We will be investigating the pros and cons of the current system by using the following methods:

* Survey: collect comments from students
* Interview: ask instructors and students in-depth of what they think of the product (talk about problem here)
* Observation: see how instructors use the product to find usability issues
* Research: review the landscape of similar e-learning solutions to make note of what could be missing from AAU Push
  + 1. System Development Tools
* Only laptops and different types of phones are required to complete this project. (Specify phones and laptop specs)
* Python is used as the language to develop the backend of the system with Django framework. (Why django?)
* HTML, CSS and Javascript are used to build frontend with several libraries and frameworks.
* React Native is a new technology out of Facebook used to make mobile apps using Javascript. We use this cross-platform framework to make Android and iOS apps.
  1. Significance of the Project

This solution will empower the academic staff to teach without limitations in communication. We will see an increased level of interaction with students.

Push will also let students focus their efforts on studying as opposed to stressing on how to get resources. It will help them stay informed and feel like they are on top of their work. This will result in boosted confidence and academic performance.

* 1. Beneficiaries

The beneficiaries of the new system are teaching staff members and their students.

* 1. Time Schedule

1. Requirement Analysis
   1. Introduction

The purpose of AAU Push is to serve as a communication platform between teachers and students in the college environment. In this regard, we will need to review each aspect of this communication.

* 1. Current System

There are two major ways teachers and students communicate in Addis Ababa University.

**Method One: Word of Mouth**

The first method is communication using a class representative. This is most common practice used by teachers to reach their students. It works in such a manner: each section of class has an assigned student that is in charge of communicating with the teacher. This student, called the class representative (Rep), talks to the teacher via phone to get announcements. Once the Rep is up to date with the latest information, he/she announces to all students in any of the following classes or, if available, in the social media group for the class. This social media group can be on Viber, WhatsApp, Telegram or other group chatting application available. In addition, if the students has a question, for example to request a change of date for an exam, the Rep will be the one to ask the teacher. When the teacher needs to give out handouts to the students, there are several methods utilized. The first and most common is giving a printout of the handout to the students to copy for themselves. The teacher could give that printout to the Rep or leave it at a photocopy shop nearby that students can go to. Another way teachers distribute class materials is by copying it to students’ flash drives. Typically the teacher would bring his/her laptop to the class and copy the material to all students with a flash drive at hand. Then students who did not have a flash would be responsible to get the files from their classmates. This method has one major issue that has been a problem for most teachers: the issue of computer viruses. As teachers plug and use multiple flash drives, they are vulnerable to attack from any virus present in these drives.

There is one other new way for distributing files: using social media applications. One of the most popular is Telegram, a social chatting application that has boomed in Ethiopia over he past two years. Teachers have been observed sending class files to Reps over this platform. And then the Reps would in turn forward these into the class groups for all to download. Or teachers create what is called a Channel on Telegram, this is similar to a group chat however only permits one user to send messages to the group and that user is the creator of the group. Teachers create a channel and let their students join. This way of communication is one-way. The teacher sends information on one side and students get it on the other.

**Method Two: AAU Push**

The second method of communication and the focus of this project is AAU Push. A website used in Addis Ababa University, and more specifically in the Department of Computer Science. Development of this website started in late 2015 by computer science students and is still maintained by them. It is currently a one-way communication platform that seeks to bridge the communication gap between teachers and students with a focus on the undergraduate program. It allows teachers to post announcements and materials for all their students at once. For this system to serve its purpose, it requires two parties to be present, all the students of the department and their instructors. Let us see in detail how this system works.

The system is first initialized. The department provides its name, course offering and number of year of the program with the sections available. Once this is fed into the system, it is ready to onboard students and teachers.

Initially, all students of the department are made to signup on the homepage of the website. There each student will enter his/her name, ID, year, section and phone number. By signing up, they are automatically subscribed to all the courses their section is taking. Once they complete signing up, they are able to login using their ID and password. Upon login, they reach the Student Dashboard.

At the Dashboard, there are a number of tabs on the menu. The first enables them to view the latest announcements. On the next tab titled ‘Courses’, all the courses a student is subscribed to at listed. Students can click on the title of the course to see all the materials that have been uploaded by their teachers. They can download the material simply by clicking on the title of the document.

The following tab in the menu is ‘Add/Drop’. Although students are automatically subscribed to all courses their section takes, it does not mean that a particular student is taking all of them in reality. Students add and drop courses from one semester to another because of many factors. To reflect this reality, students can unsubscribe a course by ‘dropping’ it in the ‘Add/Drop’ menu. Conversely, they are also able to add a course by specifying from which year and section they are taking it from. That way, announcements and materials for that course will appear in their Dashboard.

The next tab is ‘Account’ where students can update their personal and contact information. Following the ‘Account’ tab is a logout button to exit the system. This concludes the list of functionalities students receive through Push. Let us now see how Push is used by teachers.

Teachers are registered on to the website by invite only. An Excel sheet with the list of email of faculty members (or instructors) of the department, for example Department of Computer Science, is collected from the department. Once Excel file is fed into the system, Push creates users and sends out automated invitations to all instructors to join and activate their accounts. This email contains the email and temporary password that the instructor uses to login. Once an instructor open this email, they can use the credentials provided to login into the system. On their first login, he/she is asked to fill in full name and create a new password for their account. After this step they will have access to the Instructor Portal where they are able to send announcements and upload materials to their students.

Once at the Portal, instructors see a form on a tab titled ‘Post’. This form has a field to write the message of the announcement. Once instructors write their message, they pick the section they would like to send it to by ticking on the section from the list in a section titled ‘Send To’. This section holds the list of sections the teacher gives a course to. However, when teachers have just activated their accounts, there is nothing listed in the ‘Send To’ section because they have not yet specified what courses they are giving. To do this, they go to the tab titled ‘Class List’. There, they can browse through the departments year, then section in a hierarchical order then finally tick the course they teach each section from the list. This will update the ‘Send To’ on the ‘Post’ form.

To send an announcement, instructors write their message in the Post form, pick a section to send it to and click on ‘Send’. On the ‘Post’ instructors can attach or upload up to three 3 files and 2 images. The files will be listed as materials under the specific course on the student’s Dashboard.

Instructors can see the viewership of an announcement or material from the ‘Tracker’ tab. It displays the announcement message and the number of students that have seen it. The same is true for materials and its number of downloads.

Similar to the student’s Dashboard, the Portal has an ‘Account’ tab where instructors can update their personal information. Below that tab is a logout button to exit the system.

* + 1. Major Functions of the Current System / Current System Description
* Allow students to sign up on system using their ID, year and section
* Allow students to login to the system to access Student Dashboard
* Allow students to view announcements from their teachers
* Allow students to download materials sent from their teachers under each course
* Allow students to ‘add’ a course to start receiving notifications and materials on that course
* Allow students to ‘drop’ a course to stop receiving notifications and materials on that course
* Allow students to modify their personal information
* Allow teachers to signup to the system by email invitation
* Allow teachers to login to the system to access Instructors Portal
* Allow teachers to pick which section they are teaching and what course
* Allow teachers to send announcements to students for specific course they teach
* Allow teachers to upload materials to students for specific course they teach
* Allow teachers to see the number of students that have seen a specific announcement
* Allow teachers to see the number of students that have download a specific material
* Allow teachers to modify their personal information
  + 1. Problem of the Existing System

Over the time AAU Push has been in use, a number of issues have been seen. Firstly, we can see that the system is not self-contained or fully automated. For example, when teachers or students forget their passwords, the only way to reset it is by calling the administrators of the website. The admin will reset their password and give them a new one. Password recovery by email is missing from the system. Similarly, once an instructor has sent an announcement, there is not way to edit it or remove it unless by having an admin intervene. This would mean that if a teacher posts the wrong information by mistake, correcting it will not be simple.

Secondly, communication over this platform is one-way only: from teachers to students. Even though there are many cases where students would need to give an input, this is not reflected on Push. For example, feature for student assignment submission and option for asking questions is missing.

Thirdly, using AAU Push to transfer large files over the internet is difficult. Push is currently only works when teachers and students are connected to the internet. On normal cases, this should not be an issue. However, teachers often want to send files with large sizes like video or bulky textbooks. In this situation, it becomes difficult or time taking for teachers to send it. And then students will have to look for Wi-Fi to download that large file because it will be too expensive to do it via mobile data. So Push does not solve the need of sending large files in a fast and cheap manner.

Finally, the system is limited to a website. In todays fast and connected world, doing work on-the-go is demanded. For Push, mobile applications are currently not available.

* 1. Requirement Gathering
     1. Requirement Gathering Methodologies

1. User Observations: we have gathered requirements by watching teachers and students interact with the system in different use cases like sign up, login, post announcement, download materials…
2. Site Observations
3. User Interviews: we have collected requirements by interview teachers and students to find what is missing. Issues of the system and feature lists have been developed from this activity.
4. Review Other Systems: to develop feature lists, we have also researched other systems to see what needs to be added to ours.
   * 1. Results Found
5. User Observations
   1. Interface Clarity Issues: The process of posting an announcement is not clear. Users were confused where they should click.
   2. Terminology: users were confused by words used to describe actions.
6. User Interviews:
   1. Teachers suggested adding a way have students submit assignments on Push
   2. Students asked for mobile applications with push notifications
   3. Proposed System
      1. Overview

This part of the document specifies a system that can be used to solve the current systems problems that were mentioned earlier.

* + 1. Definitions

**Year**: is the number identifying the level of the student. For example, a student entering the University would have a year of 1. The following year will be 2. The limits of the year will depend on the program the student takes.

**Section**: a distinct group within a larger body of all students in a specific year. For example, a first year student in the second section would have a Section value of 2.

Course: a series of lectures or lessons in a particular subject. For example, Introduction to Computer Science

**Class**: is a course being given to a specific year and section by an Instructor. For example, Introduction to Computer Science being given to Year 1 Section 1 students is one class. The same course given to Year 1 Section 2 students is another class.

* + 1. Requirements Specification

In the proposed system, Students, Teachers (Instructors), Department Heads, Registrar Officers and the University Dean should all have accounts with their respective functions. All users should be able to recover their password through the email or phone number they provided during sign up. All accounts must be verified using the personal information provided. The new system should make the student-teacher communication a two-way street. In the same way teachers send announcements and materials, students should be able to submit assignments and ask questions. Local servers placed within the university’s network should host files sent by teachers. This is will increase availability and download speeds of course files. Whenever a student is within the university, Internet connection is not required to download files. This should feel seamless to the students. The data stored in local servers within the University should be up to date with the data on the cloud server. The system should have mobile applications and Telegram bot in sync with the website.

Students should be able to signup online by providing their ID, Name, Department, Year and Phone number. The given information should be checked against the student information list provided by the University. And if all the information provided by a student match’s with a student record in the list, that isn’t already taken, the student should be signed up and given an account. Students should login using his/her ID.

Students should be able to do the following after logging in:

First, students should be able to access announcements, files and images sent to them from their teachers, their department head, program coordinators, registrar and the university dean. To students, delivery of information from teachers is more important than delivering general announcements from the university dean, registrar or their department head. Therefore, when displaying information there should be a distinction between information from their teachers and information from their department head, program coordinators, registrar and the university dean, giving more emphasis to the first one. When students get these announcements, should be able to mark announcements as ‘Read’ to indicate that they have read it.

Second, students should be able to add classes to get information from teachers and drop classes to discontinue information from a specific class, so that the information displayed will be relevant. Students should specify the year and section of the courses they are taking to add their classes. They should be able to add only courses given to their department. Third, students should be able to access files in a simple and organized way. Files sent from teachers to students should be classified under the class they were uploaded to and files that aren’t from teachers should be classified together.

Fourth, students should be able to update their account information, but they shouldn’t be able to update their name, ID or department. Fifth, students should be able to access reminders set by their teachers, for upcoming assignments, tests and projects. Reminders should be displayed at all times when the student is logged in. Sixth, students should be able to submit assignments to their teachers. This feature should be available only for assignments that have reminders already set.

Seventh, students should be able to participate in and create discussion forums. Participants should be able to send and receive messages and images. Discussion forums should be either closed or open. Closed forums should be joined either by providing a code set by the forum creator or based on characteristics of the students, like year, section, department, courses taking, sex and campus. Every forum should have a forum ID, name and description. Eighth, students should be able to search forums by their names or forum ID. Ninth, students should be able to log out.

* + 1. Functional Requirements

1. Tracker: teachers should be able to edit their announcements
2. Reminders feature
3. Intranet Connection: Push should work offline on campus network seamlessly synced with the online database. Local servers will host large files.
4. Mobile applications: both iOS and Android for students and teachers with most functionalities
5. Assignment submission / Q&A Commenting Feature
6. Special accounts for department head, dean, and registrar
   * 1. Non-Functional Requirements
        1. User Interface and Human Factors
        2. Documentation
        3. Hardware Consideration
        4. Performance Characteristics
        5. Error Handling and Extreme Conditions
        6. Quality Issues
        7. System Modifications
        8. Physical Environment
        9. Security Issues
        10. Resource Issues
   1. System Model
      1. Scenario
      2. Use Case Model
         1. Identified Actors

|  |  |
| --- | --- |
| Actor | Description |
| Instructor | A member of staff of a department that teaches at least once course to a section |
| Student | A person who is studying at the college |
| DepartmentHead | A person in charge of leading a department, its program and staff members |
| CollegeDean | The head of a college |
| RegistrarOfficer | A person holding a position at the Registrar Office |

* + - 1. Use Case Diagram
      2. Use Case Descriptions

|  |  |
| --- | --- |
| Use case name | SendInviteToManyInstructors |
| Participating actors | Initiated by DepartmentHead  Communicates with Instructor |
| Flow of events | 1. The DepartmentHead clicks on ‘Manage Department Data’ from the Portal. 2. Push presents a form presenting two options. First is a short-text area field titled ‘Invite a Staff Member’ and second is a file upload button titled ‘Batch Invite Staff Members’. 3. The DepartmentHead uploads an Excel file containing the emails of the Instructors in the ‘Batch Invite Staff Members’ field and clicks ‘Send Invite Button’. 4. Push creates a user objects based on each email provided. Then sends a username and temporary password to each email address. |
| Entry condition | The DepartmentHead is logged in the website. |
| Exit condition | The DepartmentHead has received a confirmation of the invitation that are sent and of those that are already existent in the system. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | SendInviteToSingleInstructor |
| Participating actors | Initiated by DepartmentHead  Communicates with Instructor |
| Flow of events | 1. The DepartmentHead clicks on ‘Manage Department Data’ from the Portal. 2. Push presents a form presenting two options. First is a short-text area field titled ‘Invite a Staff Member’ and second is a file upload button titled ‘Batch Invite Staff Members’. 3. The DepartmentHead writes the email of the Instructor in the ‘Invite a Staff Member’ field and clicks ‘Send Invite Button’. 4. Push creates a user object based on that email. Then sends a username and temporary password to the email address provided. |
| Entry condition | The DepartmentHead is logged in the website. |
| Exit condition | The DepartmentHead has received a confirmation that the invitation is sent or if the email already exists in the system. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorSignUp |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The invited Instructor opens the email that presents their email for login, temporary password, and a button titled ‘Setup Account Now’. The Instructor clicks that button and is directed to [aaupush.com/login](http://aaupush.com/login). 2. Push presents a login form requesting for email and password. 3. The Instructor uses the credentials provided in the email and is logged in. 4. Push then presents a form requesting: Title, First Name, Last Name, Email, New Password and Confirm New Password. 5. The Instructor completes these form and clicks on ‘Complete Setup’. |
| Entry condition | The Instructor has accessed the invitation email and it online. |
| Exit condition | The Instructor receives a confirmation that a successful signup is done and is advanced to the Instructor Portal. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorLogin |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The Instructor clicks on login form the homepage of Push or goes loads directly to [aaupush.com/login](http://aaupush.com/login) 2. Push presents a form with two tabs ‘Student' and ‘Staff’. Under the ‘Staff’ tab are two fields, ‘Email’ and ‘Password’ and a button ‘Login’. 3. The Instructor inputs their email and password and click on ‘Login’. |
| Entry condition | The Instructor opens [aaupush.com](http://aaupush.com) |
| Exit condition | The Instructor is advanced to the Instructor Portal, OR  The Instructor is given a notice that the email and/or password is wrong. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorRecoverPassword |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The Instructor clicks on ‘Forgot Password’ link below the login form. 2. Push presents a form with ‘Email’ and a button ‘Reset Password’. 3. The Instructor inputs their email and clicks on ‘Reset Password’. 4. Push sends a password reset link to the email provided. 5. The Instructor opens that link from the email and is directed to a form containing two fields ‘New Password’ and ‘Confirm New Password’ and a button ‘Reset Password’. |
| Entry condition | The Instructor is on the login page. |
| Exit condition | The Instructor receives a confirmation and redirected to the login page. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorAttachFiles |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. From the optional fields, the Instructor clicks on ‘Attach File’ and uploads the file to be sent. 2. Push then slides open a mandatory text field title ‘Name of File’. 3. The Instructor writes the Title of the file. 4. Push then presents a field to attach another file. |
| Entry condition | The Instructor is on the ‘Post’ tab in the Portal. |
| Exit condition | The file/s is uploaded fully and the Instructor is notified. |
| Quality requirements | The Instructor has to be limited to 3 files per announcement and is notified when reached. |

|  |  |
| --- | --- |
| Use case name | InstructorSendAnnoucement |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The Instructor clicks on ‘Post’ tab in the Portal. 2. Push presents a form to the Instructor to send an announcement with mandatory fields: ‘Announcement Message’ and ‘Pick Sections’ and optional fields: ‘Attach File’ and ‘Attach Image’. 3. The Instructor fills the form by writing the message and ticking which section/s will receive it from the list given. And clicks on ‘Send Announcement’. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor receives a confirmation that the announcement is sent. |
| Quality requirements | At any point during the flow of events, this use case can includethe InstructorAttachFiles and InstructorAttachImages use cases. |

|  |  |
| --- | --- |
| Use case name | InstructorAttachImages |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. From the optional fields, the Instructor clicks on ‘Attach Image’ and uploads the file to be sent. 2. Push then presents a second field to attach another image. |
| Entry condition | The Instructor is on the ‘Post’ tab in the Portal. |
| Exit condition | The Image/s is uploaded fully and the Instructor is notified. |
| Quality requirements | The Instructor is limited to 2 images per announcement and is notified when reached. |

|  |  |
| --- | --- |
| Use case name | InstructorAddClass |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The Instructor clicks on ‘Class List’ tab in the Portal. 2. Push presents a form and a table. The form lets the Instructor browse for courses based on year and section. The table displays the current classes being taught. 3. From the Browsing Form the Instructor ticks on the courses being taught and clicks on ‘Add Class’. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor receives a confirmation that the classes are added and can now be accessed from the ‘Post’ tab. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorDropClass |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The Instructor clicks on ‘Class List’ tab in the Portal. 2. Push presents a form and a table. The form lets the Instructor browse for courses based on year and section. The table displays the current classes being taught. 3. From the Table the Instructor clicks on ‘Drop Class’ for the courses not being taught anymore. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor receives a confirmation that the classes are dropped. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorTrackAnnouncementView |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The Instructor clicks on ‘Tracker’ tab in the Portal. 2. Push presents a list of all the announcements sent by the Instructor. For each announcement, ‘Delivered To’ and ‘Read By’ fields indicate the number of students the announcement is sent to and number of students that have confirmed to have read it respectively. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor views the statistics provided. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorEditAnnouncement |
| Participating actors | Initiated by Instructor |
| Flow of events |  |
| Entry condition | The Instructor is viewing an Announcement by the InstructorTrackAnnoucnementView |
| Exit condition | The Instructor gets a confirmation that announcement has been edited. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorTrackMaterialDownload |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The Instructor clicks on ‘Tracker’ tab in the Portal. 2. Push presents a list of all the announcements sent by the Instructor. For each announcement, ‘Delivered To’ and ‘Read By’ fields indicate the number of students the announcement is sent to and number of students that have confirmed to have read it respectively. For each announcement is a ‘View More’ button. 3. The Instructor clicks on ‘View More’. 4. Push displays the files and images attached to the announcement together with the number of times the materials have been downloaded. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor views the statistics provided. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructoDownloadAssignmentSubmission |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. At the Portal, the Instructor clicks on ‘Assignments’ from the menu presented on the left. 2. The tab displays a lift of assignments that have been given by the Instructor. 3. The Instructor clicks on ‘View Submissions’ button for any of the assignments given. 4. The button opens a list submissions each one detailing the name of student, submitted assignment file, date of submission and comments left by the Student. 5. From this list, the Instructor clicks on ‘Download File’ button for each submission to review the Student’s work. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor downloads the files successfully. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorGiveAssignment |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. At the Portal, the Instructor clicks on ‘Assignments’ from the menu presented. 2. The tab displays a list of assignments that have been given by the Instructor. 3. To give a new one the Instructor clicks on ‘New Assignment’ button. 4. Push opens a form requiring the name, assignment file, deadline, which section it if for and comments related to the assignment. 5. After filling the form, the Instructor clicks on ‘Send Assignment’ button. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor gets confirmation that the assignment is successfully given. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorCreateForum |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. At the Portal, the Instructor clicks on ‘Forums’ from the menu presented. 2. The tab displays a list of active forums. 3. To give a new one the Instructor clicks on ‘Create Forum’ button. 4. Push opens a form requiring the name of the forum, description and which section/s it will include for participation. 5. After filling the form, the Instructor clicks on ‘Create Forum’ button. |
| Entry condition | The Instructor has logged into the system |
| Exit condition | The Instructor gets confirmation that the Forum is successfully created. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorDestroyForum |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The forum is displayed with the latest massages and presented with a button titled ‘Destroy Forum’. 2. The Instructor clicks ‘Destroy Forum’. |
| Entry condition | The Instructor has opened the forum by use case InstructorOpenForum |
| Exit condition | The Instructor gets confirmation that the Forum is successfully destroyed. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorOpenForum |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. At the Portal, the Instructor clicks on ‘Forums’ from the menu presented. 2. The tab displays a list of active forums. 3. The Instructor opens the forum by clicking on its name. |
| Entry condition | The Instructor is logged in the system |
| Exit condition | The Instructor gets confirmation that the Forum is successfully destroyed. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | InstructorMessageForum |
| Participating actors | Initiated by Instructor |
| Flow of events | 1. The open forum displays the latest messages and a text area and a file attachment field with a send button. 2. The Instructor writes the message in the text area, optionally attaches a file and hits ‘Send’. |
| Entry condition | The Instructor has opened the forum by use case InstructorOpenForum |
| Exit condition | The Instructor gets confirmation that the message is sent. |
| Quality requirements |  |

|  |  |
| --- | --- |
| Use case name | StudentSignUp |
| Participating actors | Initiated by Student |
| Flow of events | 1. The Student fills the sign up form on the page by inputting first name, last name, ID, Department and |
| Entry condition | The Student opens [aaupush.com](http://aaupush.com) or [aaupush.com/signup](http://aaupush.com/signup) |
| Exit condition | The Instructor gets confirmation that the message is sent. |
| Quality requirements |  |

List of Use Cases

InstructorSendAnnouncement

InstructorEditAnnouncement

InstructorAttachFiles

InstructorAttachImages

InstructorAddClass

InstructorDropClass

InstructorViewClasses

InstructorTrackAnnouncementView

InstructorTrackMaterialDownload

InstructorModifyAccountInformation

InstructorGiveAssignment

InstructorDownloadAssignmentSubmission

InstructorCreateForum

InstructorDestroyForum

InstructorOpenForum

InstructorMessageForum

StudentSignUp

StudentLogin

StudentRecoverPassword

StudentAddCourse

StudentDropCourse

StudentViewCourses

StudentViewAnnouncement

StudentReadAnnouncement

StudentDownloadMaterial

StudentModifyAccountInformation

StudentCreateForum

StudentDestroyForum

StudentOpenForum

StudentCloseForum

StudentMessageForum

SearchForForum

DepartmentHeadSendAnnouncement

RegistrarOfficerSendAnnouncement

CollegeDeanSendAnnouncement

* + 1. Sequence Diagram
    2. Activity Diagram
    3. State Chart Diagram
    4. Object Model
       1. Data Dictionary
       2. Class Modeling
       3. Dynamic Modeling
    5. User Interface