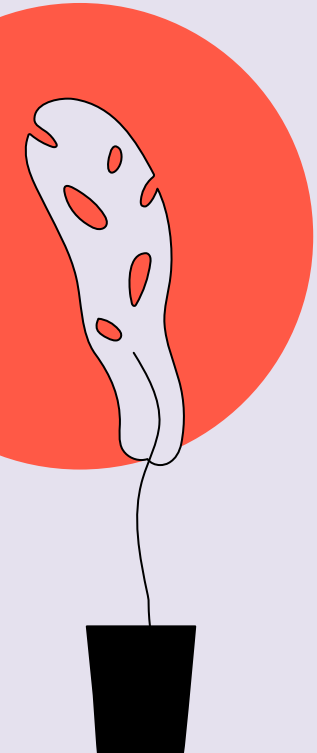
Student Affairs System

Team Members

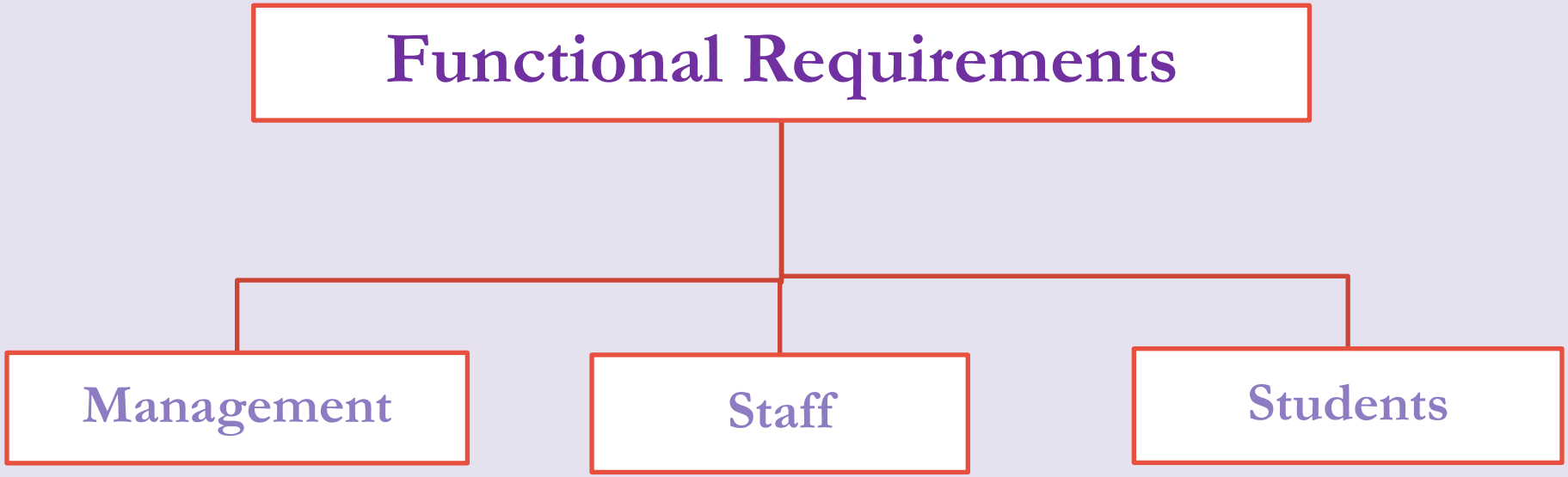


	Name
1	أسماء فتحي أبو عبد الله
2	إيمان محمد عطا الله
3	دانا أشرف الشناوي
4	صفوة فادي الريطي
5	أحمد طه العشري
6	أحمد محمد جاد
7	أحمد خالد الشرباصي
8	محمد أيمن أبو أربعة
9	محمد السيد العدوي

Requirements

Functional

Non-Functional


Functional Requirements
Management
Staff
Students

Management

Login

- Admin can login by Email
- Reset password.

Take Backup

- Manage backup for the system continuously.

Add Advertisements

- Control show or hide advertisements

Control and permission

- Add or delete or modify or update or check user's account's information.
- Give permission for each user.

Provide Technical support

- Show messages of problems and modification requests in the system from employees.

Staff

Login

- The system allows all employees to log in.
- Reset password

Enter Academic information :

- Enter the study regulations
- Enter the study programs
- Entering school teams
- Add / Edit/ Delete/ arrange courses
- Specialization

Enter Student information

- Enrollment Status
- The ability to search for a student
- Upload a photo of a student
- Tuition fees (for the band or for a particular student)
- Recruitment data
- Student documents
- Student affairs report

Assign exams

- Add exams
- Seating numbers distribution
- Distribution of examination committees
- Report on the distribution of students to the committees

Control results

- Receive Surveys from students
- Enter course grades
- Enter cumulative results:
  - ✓ total summation
  - ✓ The grand total he got
  - ✓ Total score
  - ✓ The ratio

Complaints

- Receive Emails from Students
- Send the emails to Management

Receive data

- Receive Courses that students will register
- Receive Desires that students will register

Students

Login

- The system allows the student to log in using id and password.
- Reset password.

Displays the student's data

- Personal data - previous qualification - in case of transferring to other college or institute - specialization
- Providing a university email for each student.
- Presenting a case statement for each student.

Academic registration

- Subjects
- The desires of bifurcation

Payment details

- Display payment details about expenses and books

Schedule

- The academic schedule.
- The mid-year, practical and final exams schedules.

Student progress

- The system monitors the absence of each student and warns in case of warnings
- The expected results to the student in the coming year after the result appears.

Complaints

- Providing a mailbox for the student as a technical support service within the system

Presenting the result

- Do Questionnaires For materials
- Show the result academy

Non-Functional Requirements

Operational

- Cross platforms.
- The system will be user friendly, consistent, intuitive and descriptive UI.
- The system should have a maximum of 2:3 clicks to reach any content.
- The system should support notifications.
- The system should be evolvable, scalable, testable and maintainable.
- The system should be interoperable with secondary university systems.

Performance

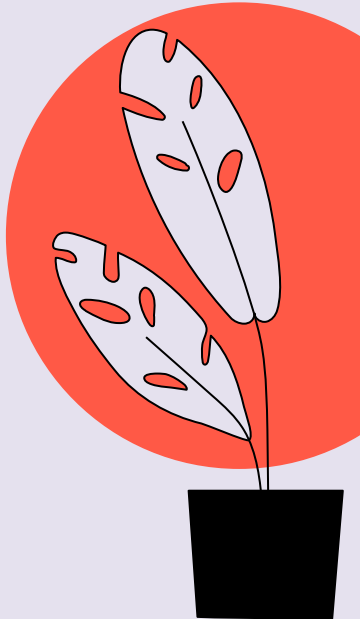
- Quick ask & response (under 2 sec).
- The system should be available 24 hours a day.
- 20 minutes are specified daily on a fixed date [3 am (low-intensity hours)] to maintain and update the system.

Security

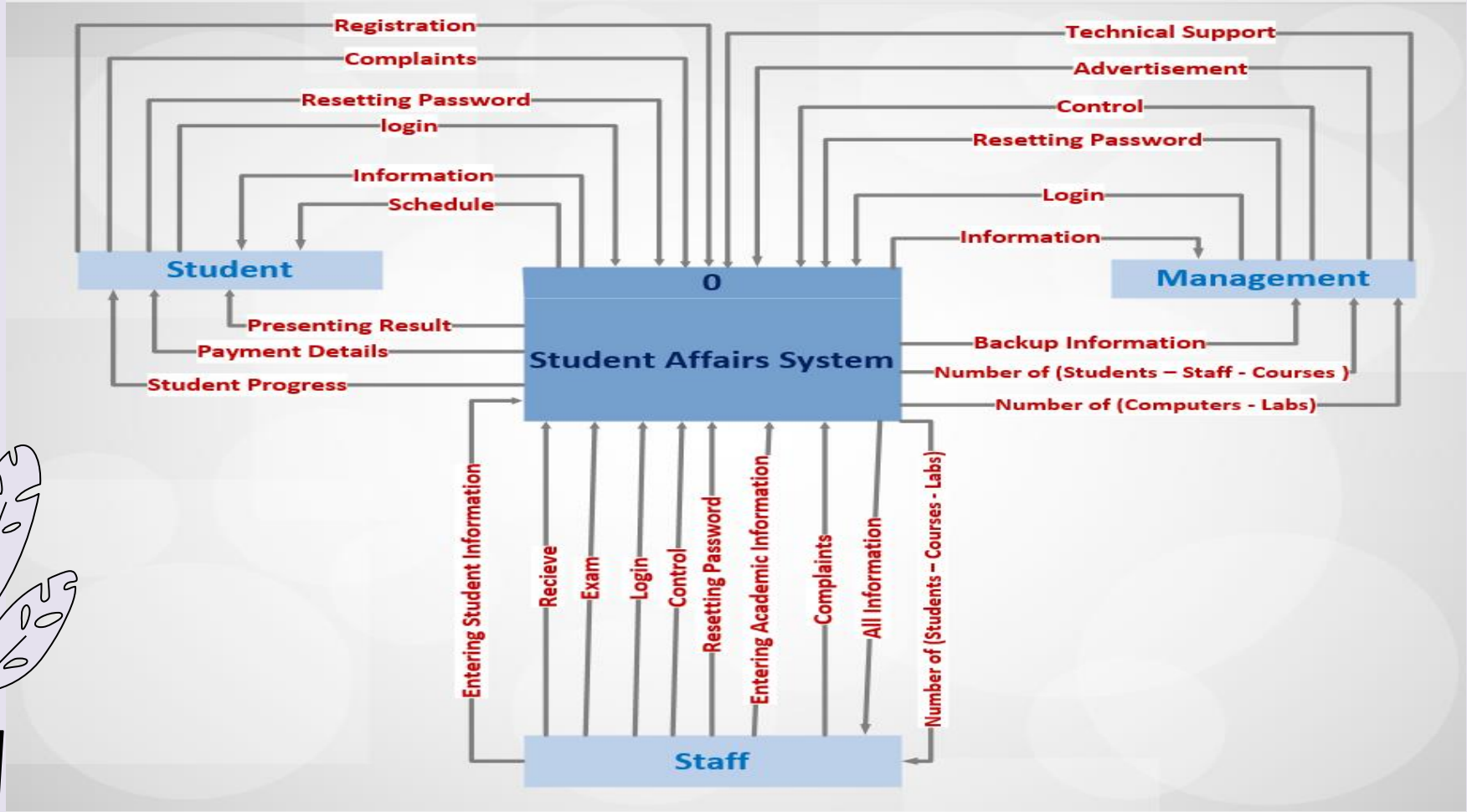
- The system must protect the user's privacy.
- Every student should have a user-customizable visibility policy for his/her personal information such as email and study timetable.
- The system will be secure and cannot affect, harm, damage the students or all users in general or their devices.

Cultural & Political

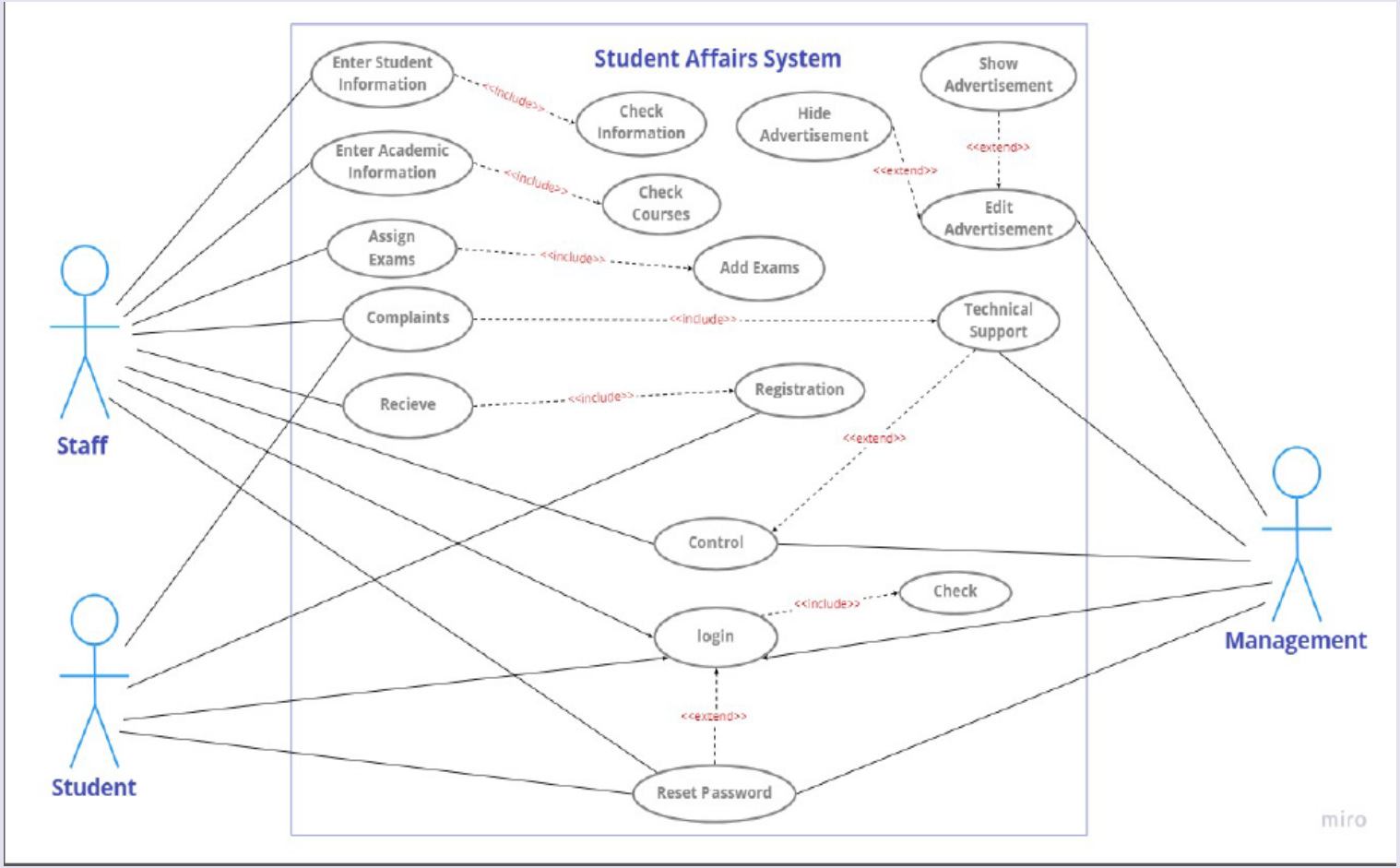
- The system should provide commonly used calendar formats.
- The system should use a related time zone.
- The system should use the currency of our country.
- The system should provide language support for all students.



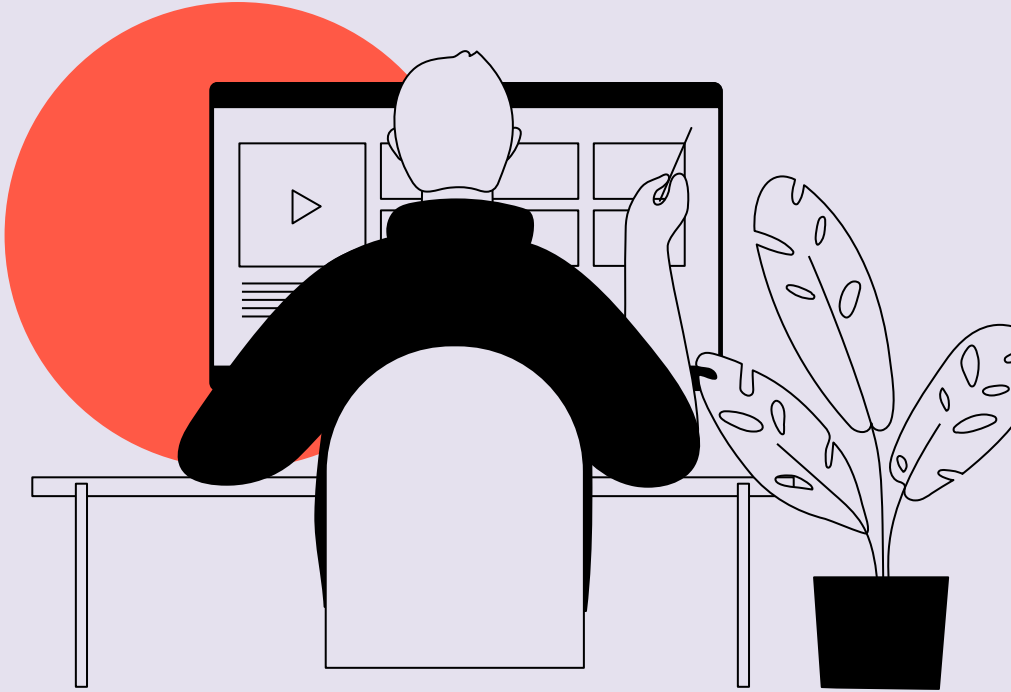
Context Diagram



Use Case Diagram



Use Case Tables



Use Case Name: **Login**	ID: **1**	Priority: **High**

Actor: **Manager, Staff, and Student**

Description: **This use case describes how a user (student, staff, or manager) logs into the Student Affair System by using their account information before they can use system.**

Trigger: **User requests to login.**

Type: ☑ **External** **Temporal**

Precondition:

1. **The user not already logged into the system.**
2. **The user has a college account.**
3. **The user is trying to log in with their account.**

Normal Course:		Information for steps:
1. **User accesses the URL**		
2. **The system prompts the user for their account credentials.**	←	**Email account**
3. **The user enters their username and password.**	←	**Password account**
4. **The system authenticates the login.**	→	**Request confirmation**
5. **The user gains access to the systems functionality.**	→	**Control functions**

Alternative:		
1. **Invalid account user or password.**	→	**Reset password**
2. **User already logged into the system before.**		

Postcondition:

1. **The user is logged in to the system.**
2. **The user has access to the functions of the system.**

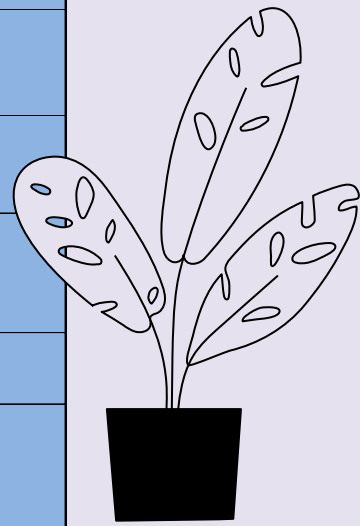
Exceptions:

1. **Incorrect password or email.**
2. **Server crash.**
3. **Power cut off.**

Summary

Inputs	Source	Outputs	Destination
Email account **Password account**	**User (student, staff, and Manager info)**	**Request confirmation** **Control functions** **Reset password**	**User (student, staff, and Manager info)** **Controlling** **reseting**

Login



Use Case Name: **Resetting Password**	ID: **2**	Priority: **High**

Actor: **User (Students, Staff) - Admin (Management).**

Description: **This use case helps the user to either set a new password or regain their old one.**

Trigger: **User clicks on "FORGOT PASSWORD?!" link.**

Type: ☑ **External** ☐ **Temporal**

Precondition:

1. **The user forgets the password.**
2. **The user can't access the system using their password.**

Normal Course:	Information for steps:
1. **The user writes the correct username and the incorrect password.**	← **Username and password.**
2. **The system shows an "invalid username or password" message error.**	→ **Error message.**
3. **The user clicks on the "FORGOT PASSWORD" link.**	
4. **The system displays the page where the user needs to answer security questions about their username and phone number.**	
5. **The user answers the questions.**	← **Answers.**
6. **The system sends an SMS code to the number for verification and prompts the user to re-enter the code.**	→ **Verification configuration.**
7. **The user enters the code.**	← **Code sent to the user.**
8. **The system displays whether the code is correct or not, if wrong the system generates a new code and sends it.**	
9. **If right, the system prompts the admin to send the user's password.**	→ **Error message.**

Alternative:

1. **Admin creates a new password and sends it through the phone number.**

Postcondition:

1. **User is logged in and able to use the system functions.**

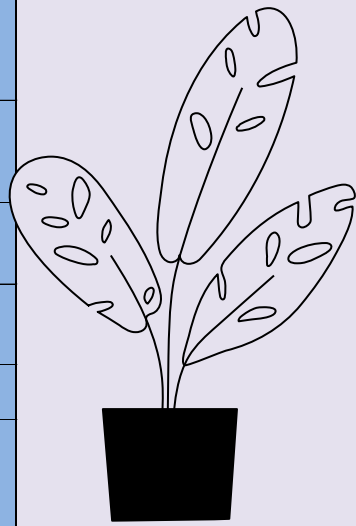
Exceptions:

1. **Server crash.**
2. **Power cut off.**

Summary

Inputs	Source	Outputs	Destination
• **Answers about phone number.** • **Verification code that was sent..**	• **User (Students and Staff info).** • **System code generator.**	• **Verification confirmation.**	• **User (Students and Staff info).**

Resetting Password



Use Case Name: **Management** Control	ID: 3	Priority: Medium

Actor: **Management**

Description: **Management** can add, delete, modify or check user's account information and give permission for each user.

Trigger: In order for the **Management** to be able to distribute the and permission modify the data.
Type: ☑**External** **Temporal**

Precondition:

- The owner must specify who has the right to put users' e-mails, or he should set e-mails according to the university's policy.
- All other administrators must have prior e-mails.
- The role of each manager must be pre-defined.

Normal Course:	Information for steps:
1.0 Modify the information for each account:	
a) Determine who is responsible for distributing emails by the owner according to the college policy, or it may be the owner himself.	← **University policy**
b) Add the data of each member of the staff.	→ **Staff information**
c) Amending the data of each member of the staff according to the developments.	→ **Staff information**
d) Modify the student data added by the staff.	→ **Students' information**
2.0 Give permissions to each user:	
a) Determine the dean and give him permissions according to the university's policy.	← **University policy**
b) Determining the members of student affairs and giving them permissions according to the university's policy.	← **University policy**
c) Determining faculty members and giving permissions according to university policy.	← **University policy**
d) Determining the permissions available to students according to the university's policy.	← **University policy**

Exceptions:

E1: System failure.

E2: A student left the college and his data was not deleted from the database.

E3:One of the employees has been promoted and his data in the database has not been modified.

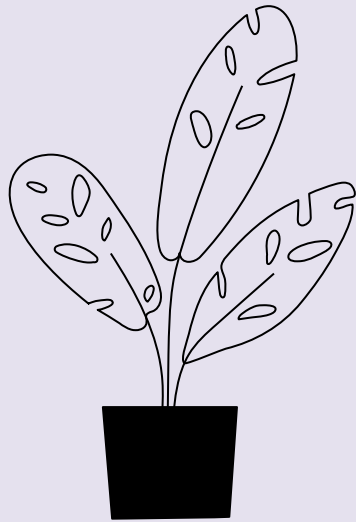
Postcondition:

1.Everyone in the working body has an email and personal information
2.Each student has an email address and personal information
3.Everyone on the system has specific powers

Summary

Inputs	Source	Outputs	Destination
University policy	controlling	students' information staff information	students' database staff database

Control



Use Case Name: **Staff** Control	ID: 3	Priority: Medium

Actor: **Staff**

Description: **The staff** can control the results and grades of the student

Trigger: So that the staff can add the student's grades.
Type: **External** ☑ **Temporal**

Precondition:

- Student absence report.
- Student's semester work grades.
- Exam scores taken by the student.
- Report on the number of exams the student did not attend.

Normal Course:	Information for steps:
1.0 Enter the student's grades:	
a. Exam grades counting.	← **Exams grades**
b. Compilation of quarterly grades	← **Student report**
c. Enter grades for each course.	→ **Grades of courses**
d. Enter total summation.	→ **Student report**
e. Enter the grand total he got.	→ **Student report**
f. Enter total score.	→ **Student report**
g. Enter the ratio.	→ **Student report**

Exceptions:

E1: System failure.

E2: Entering incorrect data as a result of a specific error, such as the similarity of names.

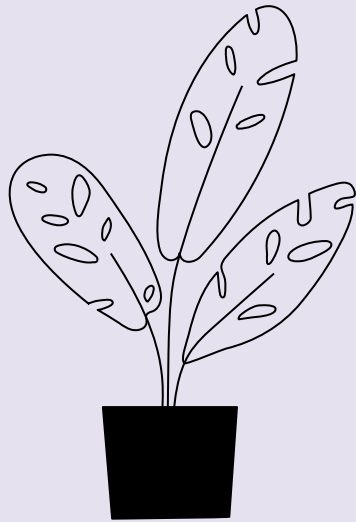
E3:More than one student forgot to enter the name on the examination paper.

Postcondition:
1.Student Status (Passed - Failed - Passed with materials).
2. A full report on the student's grades.

Summary

Inputs	Source	Outputs	Destination
Exams grades Student report	controlling controlling	full report on the student's grades.	students' database

Control



Use Case Name: **edit ads**	ID: **4**	Priority: **low**

Actor: **manager**

Description: **this use case describes adding and editing advertisement on the system**

Trigger: **announcing various events and things related to the faculty and students**

Type: **External** ☑ **Temporal**

Precondition:

1- **The web site is available**
2- **Ads are available**

.

Normal Course:	Information for steps:
1- **Add ads to the system in the place designed for them (moving bar)**	→ **New advertisement**
2- **Make any edits on the ads.**	← **Available advertisement**

Alternative:	
1. **The system sends ads via e-mail to users**	→ **New advertisement**

Postcondition:

1- **The system will display the available ads to all users dealing with the system**

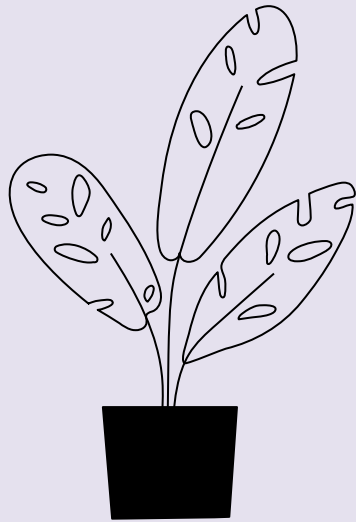
Exceptions:

1. The system failure
2. The system does not display ads that have been added
3. Ads are not available to some users

Summary

Inputs	Source	Outputs	Destination
New advertisement	**Advertisement database**	**Available advertisement**	**Advertisement database**

Edit Advertisement



Use Case Name: technical support	ID: 5	Priority: low

Actor: **management**

Description: solve staff problems.

This use case describes how a user management provide technical support for staff in the system if a problem occurs.

Trigger:

Problem occurs with staff.

Type: ☑ **External** **Temporal**

Precondition:

1- Logged in the system
2- Have e-mail.
3- Have all data of users.

Normal Course:

- **Receive the problem form staff.** ←
- **Receive message confirming from staff.** ←
- **Solve problems of the staff.** →
- **Send message confirming to staff that the problem is solved.** →

Information for steps:

Staff problems
Staff confirmations
Staff problems
Staff confirmations

Postcondition:

Problem will be solved

Exceptions:

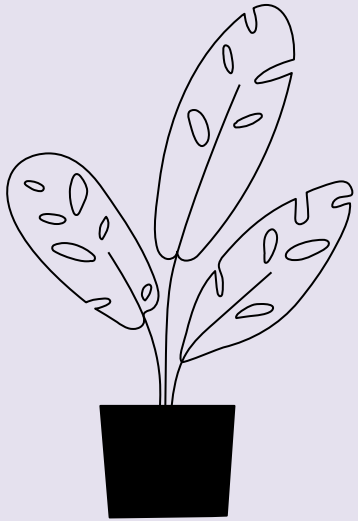
System failure

Problem still occur

Summary

Inputs	Source	Outputs	Destination
Staff problems **Staff confirmation**	**management complains database**	**Staff solved problems**	**Management complains database**

Technical Support



Use Case Name: Complaints	ID: 6	Priority: low

Actor: Staff , Student

Description: Control complaints from students
Send Problems of staff to Management

Trigger: Solve the problems of students and staff

Type: ☑ External Temporal

Precondition:

1. Problems of students and staff identified
2. Management Technical support email identified

Normal Course:	Information for steps:
a) Student	
1. Define the problem	
2. send the email of the problem to staff →	Student Problems
3. send message confirmation to staff that determine the problem is solved or not →	Student Confirmation
b) Staff	
1. Receive emails from students ←	Student Problems
2. Receive message confirming from student ←	Student Confirmation
3. problems of staff send to management →	Staff Problems
4. Send message confirming to Management that determine the problem is solved or not →	Staff Confirmation

Postcondition:

1. Problems of student are solved
2. Problems of staff are solved

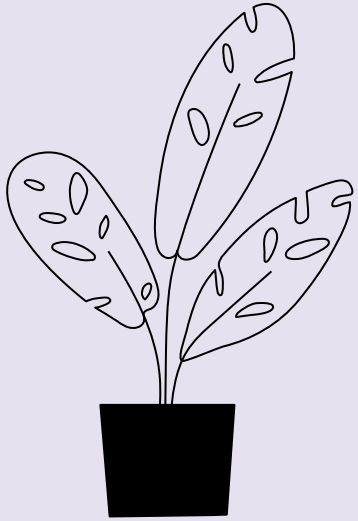
Exceptions:

1. The system falls
2. Student or staff sent a message and it did not reach

Summary

Inputs	Source	Outputs	Destination
Student Problems	Student complaints database	Student Problems	Student complaints database
Student Confirmation	Student complaints database	Student Confirmation	Student complaints database
		Staff Problems	Staff complaints database
		Staff Confirmation	Staff complaints database

Complaints



Use Case Name: Academic registration	ID: 7	Priority: High

Actor: Student

Description: The use case describes register students' courses and its desires of bifurcation.

Trigger: Students register courses and desires of bifurcation.
Type: ☑ External ☐ Temporal

Precondition:

1- The student accesses the website.
2- The student identity is authenticated.
3- Student database is available and on-line.
4- Enter student department.

Normal Course:

1.0 registering courses and desires of bifurcation.

1. The student enter courses (18 hourse)
2. The student register it's desire of bifurcation.
3. The student register the surveys.

Information for steps:

- Add courses
- Add desire of bifurcation.
- Add surveys

Alternative:

1.1 The student accesses the website.

1- The student login the system.
2- The student may register courses (12 hour or 21 hour).
3- The student only checks the accuracy of the information.

- Account information
- Courses
- Information

Postcondition:

1. All academic registration is appeared for students after entrance to academic registration department.
2. Student submit changes after he finish.

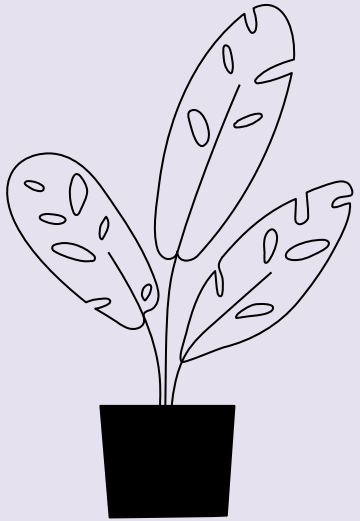
Exceptions:

1- Invalid password or email.
2- Student database is not available and off-line.
3- The student is trying to modify data that he does not have permission to modify.
4- The student may enter courses has more hours than expected.
5- Internet disconnection

Summary

Inputs	Source	Outputs	Destination
1. Add courses 2. Add desire of bifurcation 3. Add surveys 4. Account information 5.Courses	1. Students database. 2. Academic registration department.	Information	Registration database.

Registration



Use Case Name: **Receive**	ID: **8**	Priority: high

Actor: **Staff**

Description: **Receive Courses that students will register**
Receive Desires that students will register

Trigger: **When Student Register Courses**
When Student Register Desires

Type: **External** ☑ **Temporal**

Precondition:
1. **Courses that Student register are available**
2. **Desires that Student register are available**

Normal Course:	Information for steps:
1. **Receive Courses** ←	**Courses**
2. **Receive Desires** ←	**Desires**

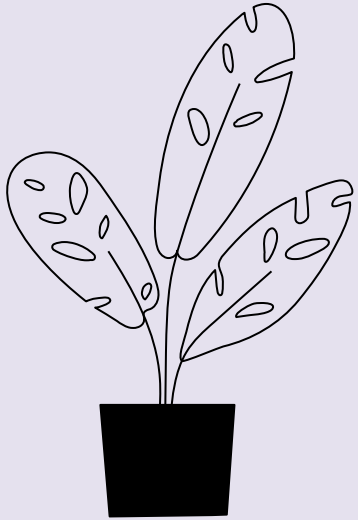
Postcondition:
1. **Have a report of courses that student register**
2. **Have a report of Desires that student register**

Exceptions:
1. **The system falls**
2. **Did not receive Courses**
3. **Did not receive Desires**

Summary

Inputs	Source	Outputs	Destination
Courses **Desires**	**Students Courses** **Students Desires**	**Report of courses and Desires**	**Student database**

Receive



Use Case Name: Assign Exam	ID: 9	Priority: high

Actor: **Staff**

Description: **Add exams to the system, Seating numbers distribution & Distribution of examination committees**

Trigger:

1- **Determining the exam for each course**
2- **Determining** Seating numbers
3- **Determining** Examination committees

Type: **External** ☑ **Temporal**

Precondition:

1- Sign in successfully
2- Exams to be added are available
3- Seating numbers to be added are available
4- Examination committees to be added are available

Normal Course:

1- Entry to the department
2- Entry to the course
3- **Add Exams**
4- **Seating number distribution**
5- **Distribution of examination committees**
6- Make a report on the distribution of students to the committees

Information for steps:

1- Department Name
2- Course Name
3- Exams
4- Seating number
5- Examination committees
6- Report on the distribution **of students to the committees**

Postcondition:

1- The system will display the exams
2- The system will display the Seating number
3- The system will display Examination committees
4- The system will display the Report on the distribution

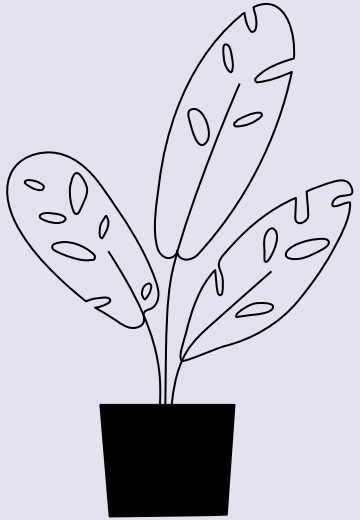
Exceptions:

1- The system failure
2- The system does not display the exams
3- There was an error in the exams
4- The system does not display the Seating number
5- The system does not display Examination committees
6- The system will display the Report on the distribution

Summary

Inputs	Source	Outputs	Destination
1- Department Name 2- Course Name	1- **Department Database** 2- **Course Database**	1- **Exams** 2- **Seating number** 3- Report on the distribution **of students to the committees**	1- **Department Database** 2- **Course Database**

Assign Exams



Use Case Name: Academic registration	ID: 7	Priority: High

Actor: Student

Description: The use case describes register students' courses and its desires of bifurcation.

Trigger: Students register courses and desires of bifurcation.

Type: ☑ External ☐ Temporal

Precondition:

1- The student accesses the website.
2- The student identity is authenticated.
3- Student database is available and on-line.
4- Enter student department.

Normal Course:	Information for steps:
1.0 registering courses and desires of bifurcation.	
1. The student enter courses (18 hourse)	← Add courses
2. The student register it's desire of bifurcation.	← Add desire of bifurcation.
3. The student register the surveys.	← Add surveys

Alternative:	
1.1 The student accesses the website.	
1- The student login the system.	← Account information
2- The student may register courses (12 hour or 21 hour).	← Courses
3- The student only checks the accuracy of the information.	→ Information

Postcondition:

1. All academic registration is appeared for students after entrance to academic registration department.
2. Student submit changes after he finish.

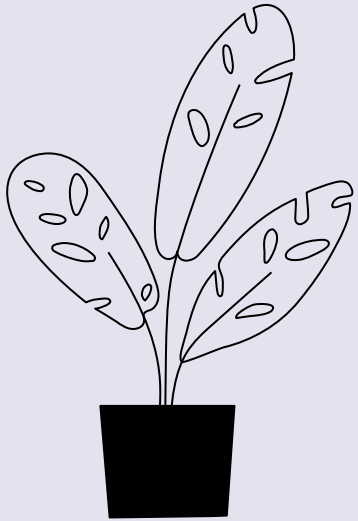
Exceptions:

1- Invalid password or email.
2- Student database is not available and off-line.
3- The student is trying to modify data that he does not have permission to modify.
4- The student may enter courses has more hours than expected.
5- Internet disconnection

Summary

Inputs	Source	Outputs	Destination
1. Add courses 2. Add desire of bifurcation 3. Add surveys 4. Account information 5.Courses	1. Students database. 2. Academic registration department.	Information	Registration database.

Enter Academic Information



Use Case Name: **Enter Student Information**	ID: **11**	Priority: **High**

Actor: Staff

Description: Adding Student Information to the system

Trigger: View data about each student

Type: ☑ **External** **Temporal**

Precondition:

1- **Availability of data to be added**

Normal Course:	Information for steps:
1- Add Enrollment Status 2- Upload a photo of a student 3- Add Tuition Fees 4- Add Recruitment data 5- Add student documents 6- Add Student affairs report	1- Enrollment Status 2- Upload Photo 3- Tuition Fees 4- Recruitment Data 5- Student documents 6- Student affairs report

Postcondition:

1 – The system will display all the student information

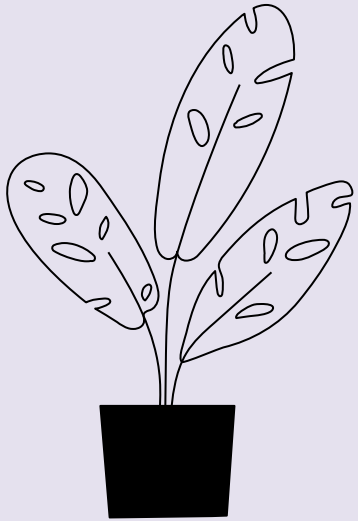
Exceptions:

1- The System failure
2- The system does not display the student information
3- There was an error in the student information

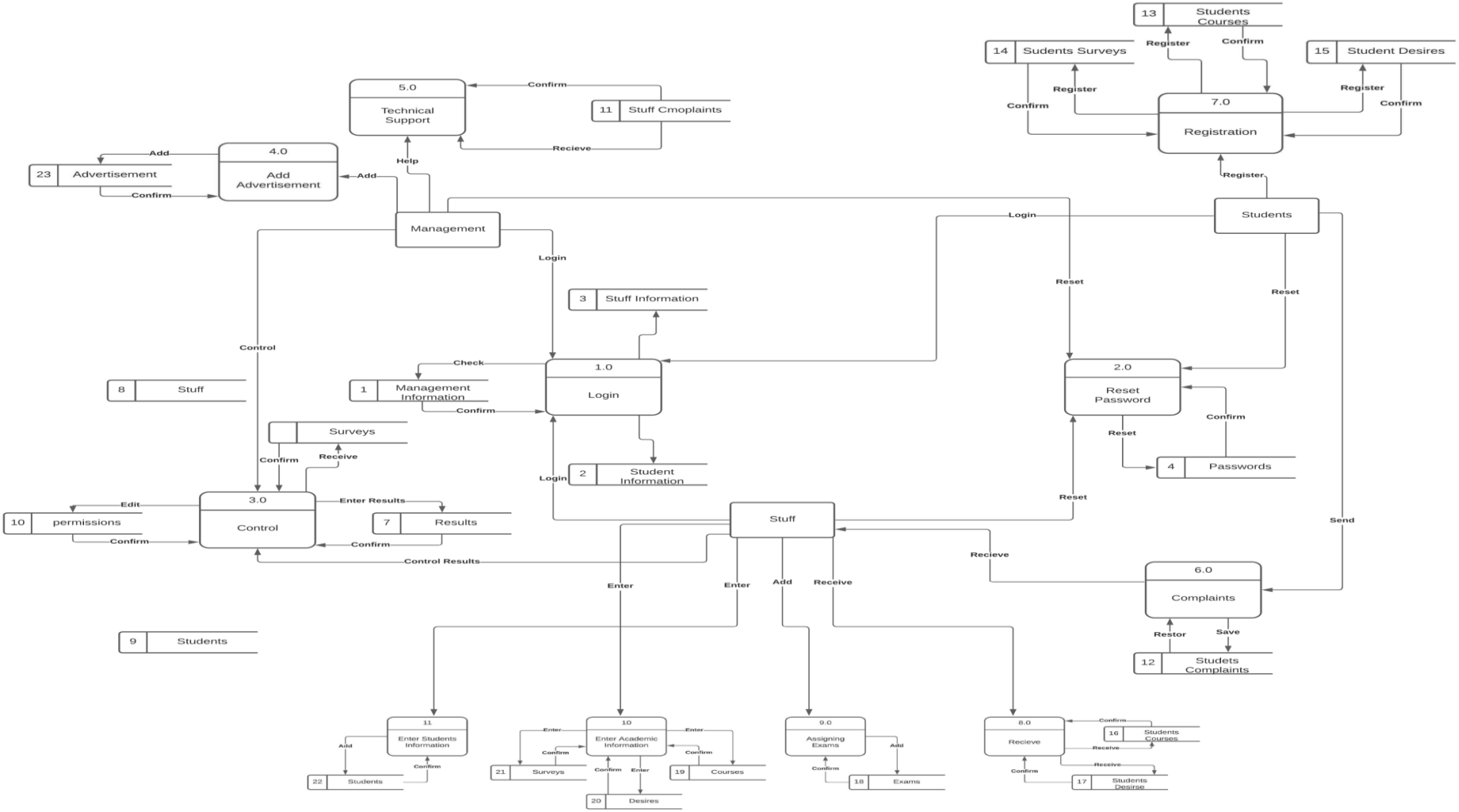
Summary

Inputs	Source	Outputs	Destination
7- Enrollment Status 8- Upload Photo 9- Tuition Fees 10- Recruitment Data 11- Student documents 12- Student affairs report	**Students Database**	1- **All student's information** 2- **Students affairs report**	**Students Database**

Enter Student Information



Data Flow Diagram


13 Students Courses
14 Sudents Surveys
15 Student Desires
Register
Confirm
Register
Confirm
Register
Confirm
7.0
Registration
Register
5.0
Technical Support
Confirm
11 Stuff Cmoplaints
Recieve
Help
4.0
Add Advertisement
Add
Add
23 Advertisement
Confirm
Management
Students
Login
Login
Reset
Reset
3 Stuff Information
Control
Check
1.0
Login
8 Stuff
1 Management Information
Confirm
2.0
Reset Password
Confirm
Reset
Surveys
Confirm
Receive
4 Passwords
Login
2 Student Information
Edit
3.0
Control
Enter Results
10 permissions
Confirm
7 Results
Confirm
Reset
Stuff
Send
Control Results
Recieve
6.0
Complaints
Enter
Enter
Add
Receive
Restor
Save
9 Students
12 Studets Complaints
11
Enter Students Information
Add
Confirm
22 Students
10
Enter Academic Information
Enter
Confirm
Enter
Confirm
21 Surveys
Confirm
Enter
19 Courses
20 Desires
9.0
Assigning Exams
Add
Confirm
18 Exams
8.0
Recieve
Confirm
16 Students Courses
Receive
Receive
Confirm
17 Students Desirse