***SCENARIOS***

*1.*

|  |  |
| --- | --- |
| *Scenario name* | Instructor Selects Course -Mertali & Gülden |
| *Participating actor*  *Instances* | *Mert:Instructor*  *OR*  *Deniz: Assistant* |
| *Flow of events* | 1. Mert logs in the application called ‘AW SHEET’ by her own mobile Phone. 2. Mert opens the Courses tab by touching that icon and tapps “Select Courses” functionality to select a course to add on her schedule. 3. He selects the course and enters the course details, including numbers of midterms & quizzes, grading policy, assistant of course and attendance requirements. 4. He taps confirmation button. |

*2.*

|  |  |
| --- | --- |
| *Scenario name* | *Instructor/Assistant Registration -Mertali & Gülden* |
| *Participating actor*  *Instances* | *Burak:Admin* |
| *Flow of events* | 1. *Burak, the Admin,* enters the system by using his own   username and password on ‘AW SHEET’.   1. He tapped the button called “Add a New Instructor” that is under the “Add A New Member” tab. 2. Burak , fills up the information of instructor-The informations are name, surname, email address and her/his department name- Then taps the confirmation button. 3. System checks the information of instructor that satisfies the constraints of system, then adds the instructor on database and sends an email that contains her/his username and password with respectively. 4. Afterall, Burak, the Admin, gets notification that the instructor has been added to the database successfully. |

*3.*

|  |  |
| --- | --- |
| *Scenario name* | *Student Withdraws Course -Mertali* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner logs in the application called ‘AW SHEET’ by her own mobile phone with his personal password. 2. He taps the “*Courses*” Tab, then selects the course that he wants to withdraw.. 3. In the “Course” menu, there is “more” icon. That icon is nothing but the simple drop-down menu. 4. He taps the “Withdraw Course” button on drop-down list. 5. Caner confirms the alert box that asks him if he is sure about he wants to withdraw the course,that appears right after he taps the “Withdraw Course”. 6. After the confirmation of alert box, the feedback comes to Caner saying that “He successfully withdrawn the course." |

*4.*

|  |  |
| --- | --- |
| *Scenario name* | *Log-in Operation -Mertali* |
| *Participating actor*  *Instances* | *Caner:Student OR Mert:Instructor*  *OR Burak:Admin* |
| *Flow of events* | 1. Caner connects to the AW-SHEET application via his mobile phone. 2. Caner fills the his own login information to the Login-Form(i.e. [name.surname@isik.edu.tr](mailto:name.surname@isik.edu.tr) ,123123) 3. He taps the ‘Login’ button. |

*5.*

|  |  |
| --- | --- |
| *Scenario name* | Student Course Addition -Mertali |
| *Participating actor*  *Instances* | *Caner:Student* |
| *Flow of events* | 1. Caner logs in the application called ‘AW SHEET’ by her own mobile Phone.Caner taps the ‘Course’ button. 2. Then, Caner taps “Add Course” button.”Add Course” button is a function to add a course from course table to his schedule. 3. Caner, the Student, types the name of the course in search box and selects the course that he is taking at the that semester. 4. Caner taps `Add Selected Courses` to invoke add that course to his schedule. |

*Efem ?.?*

*Çok tatlı benim olsunmu oley   
 ¡ \_\_¡*

*[O.o]*

*/)\_\_)*

*-”--”---(Guldenindir dokunan ölür ..!)*

*6.*

|  |  |
| --- | --- |
| *Scenario name* | Instructor Edits Course Details -Mertali |
| *Participating actor*  *Instances* | *Mert:Instructor* |
| *Flow of events* | 1. Mert logs in the application called ‘AW SHEET’ by her own mobile Phone.Mert taps the ‘Course’ button. 2. Then Mert taps Courses icon on tab item. 3. The Instructor,selects the course on list then course page appears. 4. He,the Instructor touches the option icon and taps “Edit Course” 5. The Course Form appears,the instructor selects the item that he want to change(Allowed items are , Course syllabus , # of quizzes and midterms e.t.c) the course that he is giving.   5.1.Mert,the Instructor types name of the course in search box and selects the course that he is giving at the that semester.   1. Mert , types the form item and changes it , that he selected. |

*7.*

|  |  |
| --- | --- |
| *Scenario name* | Admin Adds Course List -Mertali |
| *Participating actor*  *Instances* | *Burak:Admin* |
| *Flow of events* | 1. Burak, connects the application by her own mobile phone. Burak ,the *admin,* fills up the ‘Login-Form’ by using her own email and password.He taps login button. 2. Burak,the admin, taps the Courses tab. 3. In courses tab burak touches the Choose a file button to upload whole courses by .exel document.   3.1. In Courses tab he is able to add courses by hand with using ‘Add a Course’ button to insert course that is considering to give in that semester.  3.2.After tapping ‘Add a Course’ button he actually fills the mini-form that is contains the course name and course code.   1. Burak, taps the Submit button to invoke addition action. 2. Then alert box appears and asks the *admin,* If he is sure to add that course 3. Burak,the admin, confirms that “Alert-box” by tapping ‘✓’ icon on his request. 4. Then Feedback appears to the Burak. |

*8.*

|  |  |
| --- | --- |
| *Scenario name* | Admin determines Add/Drop-Mertali |
| *Participating actor*  *Instances* | *Burak:Admin* |
| *Flow of events* | 1. Burak logs in the application called ‘AW SHEET’ by her own mobile Phone.Burak taps the ‘Course’ button. 2. Then Burak taps , ‘Course’ Tab. There is a button called ”Set Add & Drop Date ” is a button to set add & drop date. 3. Burak,the Instructor, tapps it and sees a mini-form that asks for interval Dates. 4. He enters the dates that is allowed.(Exception Handling !) 5. Burak taps `Done` to invoke set date of add & drop week to that semester. |

9.

|  |  |
| --- | --- |
| *Scenario name* | *Instructor Takes Attendances by Swipe Function -Gülden* |
| *Participating actor*  *Instances* | *Mert: Instructor OR*  *Deniz: Assistant*  *Caner: Student* |
| *Flow of events* | 1. *Mert*, the instructor, is in the class, before the lecture, he logs in the application called ‘AW SHEET’ by his own mobile Phone aiming to take attendance of the class. 2. He selects the course, from the courses he is giving. 3. He taps the “Take Attendance” button. He chooses “Take attendance by swipe” function, by tapping a button. 4. Students start to appear on screen, one by one, respectively to the student list. For present students, *Mert* swaps the screen to right and for absent students, *Mert* swaps to left. 5. After, he swipes a student, Caner, as absent, by mistake, *Mert* swipes up, he sees the previous student marked, which is Caner. He corrects by marking again. 6. He finishes taking attendance of the class. He confirms the attendance and the attendance list is saved to the system. |

*10.*

|  |  |
| --- | --- |
| *Scenario name* | *Instructor Takes Attendances by QR Code Function -Gülden* |
| *Participating actor*  *Instances* | *Mert: Instructor OR*  *Deniz: Assistant* |
| *Flow of events* | 1. Mert, the instructor, is in the class, before the lecture, he logs in the application called ‘AW SHEET’ by his own mobile Phone aiming to take attendance of the class. 2. He selects the course, from the courses tab he is giving. 3. He taps the “Take Attendance” button and then chooses “Take attendance by QR Code” function, by tapping a button. 4. A QR code appears on screen. Students join to attendance. 5. After Mert takes the attendance, he confirms the attendance. 6. Attendance is saved to the system. |

*11.*

|  |  |
| --- | --- |
| *Scenario name* | *Instructor Takes Attendances by Table -Gülden* |
| *Participating actor*  *Instances* | *Mert: Instructor OR*  *Deniz: Assistant* |
| *Flow of events* | 1. Mert, the instructor, is in the class, before the lecture, he logs in the application called ‘AW SHEET’ by his own mobile Phone aiming to take attendance of the class. 2. He selects the course, from the courses tab he is giving. 3. He taps the “Take Attendance” button and then chooses “Take attendance by Table” function, by tapping a button. 4. List of students appear on screen. 5. Mert takes attendance by filling the check boxes for each student in the class. 6. After Mert takes the attendance, he confirms the attendance. 7. Attendance is saved to the system. |

*12.*

|  |  |
| --- | --- |
| *Scenario name* | *Student Scans QR Code -Gülden* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Berk launches the “AW SHEET” application using his phone. 2. He logs in using his e-mail and password. 3. He taps the camera/scan button. 4. He scans the QR Code displayed in the instructor’s phone. 5. He is included in the attendance. |

*13.*

|  |  |
| --- | --- |
| *Scenario name* | *Enter Grade -Gülden&Berk* |
| *Participating actor*  *Instances* | *Mert: Instructor OR*  *Deniz: Assistant* |
| *Flow of events* | 1. Mert, the instructor logs in the application called ‘AW SHEET’ by his own mobile phone. 2. He selects the course from the courses tab he is giving. 3. He taps to a exam registered before. 4. He taps “Enter Grade for exam” button. 5. List of students appear on screen. 6. He enters the grades one by one for each student. 7. Mert checks the list before publishing it, he taps to save button. 8. When he publishes, the system sends a push notification to every student. |

14.

|  |  |
| --- | --- |
| *Scenario name* | *Student Registration -Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner, the student opens the “AW SHEET” app. He taps “Sign-Up“ button and fills up the informations needed. Informations needed are; Name, Surname, Student Number, Email and Password. Than taps the “Confirmation” button. 2. An e-mail is sent to the Caner’s email to confirm that Caner is a student. 3. Caner checks his email than taps the “confirm account” link in his email. 4. After Caner confirms his e-mail, system adds Caner as a student to database. 5. Caner logs in to system by tapping “Login” button after he fills the “Email” and “Password” text-box. |

*15.*

|  |  |
| --- | --- |
| *Scenario name* | *Student Tracks Grade -Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner, the student logs-in to the “AW SHEET” app. 2. He taps the “Courses” button. The list of courses that he is taking opens. 3. He taps the “more (. . .)” icon for any course 4. A drop-down list appears and he taps the “Check Grades” button. 5. The grades of Caner from the selected course is listed in a pop-up tab. |

*16.*

|  |  |
| --- | --- |
| *Scenario name* | *Student Tracks Attendance -Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner, the student logs-in to the “AW SHEET” app. 2. He taps the “Courses” button. The list of courses that he is taking opens. 3. He taps the “more (. . .)” icon for any course 4. A drop-down list appears and he taps the “Check Attendance” button. 5. The attendance records of Caner from selected course is turned to a percentage value and shown to Caner in a pop-up tab. |

*17.*

|  |  |
| --- | --- |
| *Scenario name* | Student Receives Announcement *-Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner, the student logs-in to the “AW SHEET” app. 2. *On the main page he taps the “Notifications” button.* 3. *Announcements or material given by Instructor or Assistant is listed in the “Notifications” page. Only the lecture name and a short preview of the announcement is shown.* 4. *He taps any of them to view it fully.* 5. *After he is done checking announcements, he closes the app.* |

*18.*

|  |  |
| --- | --- |
| *Scenario name* | Student Calculates Grade *-Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner, the student logs-in to the “AW SHEET” app. 2. He taps the “Courses” button. The list of courses that he is taking opens. 3. He taps the “more (. . .)” icon for any course 4. A drop down list appears and he taps the “Calculate Grade” button 5. A new tab is openedwith some text boxes. Which are Midterms, Homeworks, Quizzes, Attendance points. Numbers of these text boxes depend on the syllabus of the selected course. 6. *Caner enters his predicted grades to given text boxes.* 7. *Predicted final grade is shown as result.* |

*19.*

|  |  |
| --- | --- |
| *Scenario name* | Student Resets Password *-Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner launches and logs-in to “AWSHEET”. 2. He taps the “Settings” button on home page. 3. From the Settings list, Caner taps the “Password Settings” 4. He is asked to fill Current Password and New Password. 5. If Current Password is correct and not the same with New Password password is changed. 6. An alert box appears that states the successfulness of the action. |

*20.*

|  |  |
| --- | --- |
| *Scenario name* | Student Forgets Password *-Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner launches the app “AWSHEET”. 2. He taps the “Forgot Password” button on login page. 3. From the new tab opened he is asked to fill email of his existing account. 4. After filling the information needed Caner taps the “Send New Password” button. An alert box appears that states the successfulness of the action. 5. A new password is sent to given via e-mail. |

*21.*

|  |  |
| --- | --- |
| *Scenario name* | *Assistant Sends Quiz Grades To Instructor -Berk* |
| *Participating actor*  *Instances* | *Deniz:Assistant*  *Mert:Instructor* |
| *Flow of events* | 1. Mert, the instructor launches the “AW Sheet” application.He want to check quiz grades which has been done by assistant Deniz. Mert opens courses tab, click on the course and tap on grade request from the assistant. 2. Deniz gets a notification which has information about instructor request.The information consist of course name, course code and quiz number. 3. Deniz sees the request and its possible send option.Deniz uses list method to fill quiz grades of students. 4. After filling, she sends them by tapping button which appears at the end of grade store procedure. 5. Mert gets a notification when quiz grades has returned. Mert will launch the app by tapping notification and will have two options: one of them is publishing and other one is rejecting and review option. Mert selects reject button and app redirect Mert to email app to send an email about false and the quiz grades request will be resend. |

*22.*

|  |  |
| --- | --- |
| *Scenario name* | *Instructor Gets Attendances on LAB/PS -Berk* |
| *Participating actor*  *Instances* | *Mert:Instructor* |
| *Flow of events* | 1. Mert, the instructor launches the “AW Sheet” application.He want to check attendance information which has been done on lab/ps sections. Mert opens courses tab, click on the course and tap on attendance request from the assistant. 2. The system sends attendance lists to instructor.And merge them with course attendances. 3. Mert get two option for publishing now or waiting until a date option. 4. Mert selects publish now and attandances will be visible to students. |

*23.*

|  |  |
| --- | --- |
| *Scenario name* | Student checks his/her schedule. *-Korhan* |
| *Participating actor*  *Instances* | *Caner: Student* |
| *Flow of events* | 1. Caner launches and logs in to the app “AWSHEET”. 2. On the Homepage Caner is given his weekly schedule that shows the course names and classrooms of the courses that he is taking. 3. He then taps the “Schedule” button on the home page to view his full calendar that shows midterm dates and homework/project due dates. |

*24.*

|  |  |
| --- | --- |
| *Scenario name* | *Admin Deletes Instructor -Mertali & Gülden* |
| *Participating actor*  *Instances* | *Burak:Admin* |
| *Flow of events* | 1. *Burak, the Admin,* enters the system by using his own username and password on ‘AW SHEET’. 2. Burak,the Instructor, enters the “User” tab by touching icon of it. 3. He taps the button called “Delete Instructor” that is under the “Delete Member” tab. 4. Burak, the admin, types name of the instructor in search box and selects the instructor. 5. Burak, taps the “Delete” button to invoke removing member action. 6. Then alert box appears and asks the *admin,* If he is sure to remove that Instructor is selected. 7. Burak,the admin, confirms that “Alert-box” by tapping ‘✓’ icon on his request. 8. Afterall, Burak, the Admin, gets notification that the instructor has been deleted from the database successfully. |

***USE CASES***

***1.***

|  |  |
| --- | --- |
| *Use case name* | *Register -Korhan* |
| *Participating actors* | *Requested by Student*  *Interacts with AWSHEET* |
| *Flow of events* | 1. Student taps to “Sign-up” button. 2. Student fills up the form by entering Name, Surname, Student Number, Email and Password. Than taps the “Confirmation” button. ***[Time Out]*** 3. *AWSHEET sends a confirmation email to the email address given by Student.*   ***[Invalid Email]***   1. *Student confirms the account by tapping the link in confirmation mail.* 2. *AWSHEET adds Student to database as a user.*      1. *Student successfully logs in to his/her account after filing “Email” and “Password” text boxes.* |
| *Entry Conditions* | * *Student downloads and opens the app “AW SHEET”* |
| *Exit Condition* | * *Student successfully logs in to his/her account. OR* * *Doesn’t receive the confirmation e-mail. OR* * *Student stays idle for a determined time and Time Out happens.* |
| *Quality Requirement* | * *Confirmation email is sent to given email address within 2 minutes.* |

*Exceptional Cases:*

**[Invalid Email] :** If the email given by student is wrong or he isn’t the owner of given email, Student can’t confirm his/her account. Student needs to retry the process.

**[Time Out] :** If Student stays idle for a determined time period system stops the running function.

2.

|  |  |
| --- | --- |
| *Use case name* | Login - Berk |
| *Participating actor* | Initiated by Student OR  Instructor OR  Assistant  Interacts with AW Sheet |
| *Flow of events* | 1. AW Sheet displays login and register buttons and email and password fields on screen. 2. Student fill the form and click login button. 3. AW Sheet takes the data and checks their validity.**[Invalid Email][Invalid password][More Than 3 Invalid Try]** |
| *Entry condition* | * Student downloads and launches “AW Sheet” |
| *Exit condition* | * Student successfully logins to the system. |

*Exceptional Cases*

*[Invalid Email] : If email the student entered is wrong or not registered.The system gives an warning message and returns to login screen again.*

*[Invalid Password] : If password the student entered is wrong.The system raised an warning and return login screen with the email field is filled.*

*[More Than 3 Invalid Try]:If student enters email or password invalid more than 3 times, recaptcha will appear and user must enter recaptcha every time he/she try to login.*

3.

|  |  |
| --- | --- |
| *Use case name* | Logout -Berk |
| *Participating actor* | Initiated by Student OR  Instructor OR  Assistant  Interacts with AW Sheet |
| *Flow of events* | 1. Student taps on the wheel icon on the top right. 2. From drop down menu selects logout selection. |
| *Entry condition* | * Student logged in successfully. * Student launches “OW Sheet”. |
| *Exit condition* | * AW Sheet redirect student to the login page. |

4.

|  |  |
| --- | --- |
| *Use case name* | Reset Password -Berk |
| *Participating actor* | Initiated by Student OR  Instructor OR  Assistant  Interacts with AW Sheet |
| *Flow of events* | 1. Student launches the app. 2. Student tap on wheel on top right of the screen. 3. Select password settings option. 4. Enters old password and new password two times.[Invalid Password][Unsecured Password]Taps on change password. 5. OW Sheet change password of student.Raise an alert that contains information about successfulness of action. 6. Students receives the alert. |
| *Entry condition* | * Student logged in successfully. * Student launches “OW Sheet”. |
| *Exit condition* | * Clicking on change password button. |

*Exceptional Cases*

*[Unsecured Password] : If the new password has less than eight characters, the system alerts student and redirects to change password page.*

*[Invalid Password] : If the old password haven’t match with student’s password it alerts the user and redirect to change password page.*

[*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.

[TimeOut]:This appears when the user does not do anything within a time that has already setted before.

5.//

6.

|  |  |
| --- | --- |
| *Use case name* | DeleteUser -Mertali |
| *Participating actor* | Initiated by Admin |
| *Flow of events* | 1. *The Admin,* enters the system by using his own username and password on ‘AW SHEET’. 2. Admin taps the button called “Remove a Instructor” that is under the “Member” tab.   2.1.Admin taps the button called “Remove a Assistant” that is under the “ Member” tab.  2.2.Admin taps the button called “Remove a Admin” that is under the “Member” tab.   1. The Admin selects the Instructor on form with using drop down menu and then selects the reason of the removing of Instructor then Admin submits the form with tapping on confirmation button.   3.1. The Admin selects the Assistant on form with using drop down menu and then selects the reason of the removing of Instructor.   1. Admin submits the form with tapping on confirmation button. 2. AW Sheet displays an alert box and asks the *admin,* If s/he is sure to remove that Instructor is selected. 3. The admin confirms that “Alert-box” by tapping ‘✓’ icon on his request. 4. Afterall, the Admin, gets notification that the instructor has been deleted from the database successfully. |
| *Entry condition* | User is logged in to system as Admin |
| *Exit condition* | Instructor/Assistant/Admin added to the Aw Sheet. |
| *Exceptional Cases :* | *[removingUser]:*It would basically be threwen when the admin tries to remove a user from DB that does not exist.  [*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.  [TimeOut]:This appears when the user does not do anything within a time that has already setted before. |

7.

|  |  |
| --- | --- |
| *Use case name* | AddUser -Mertali |
| *Participating actor* | Initiated by Admin  Interacts with AW Sheet |
| *Flow of events* | 1. *The Admin,* enters the system by using his own username and password on ‘AW SHEET’. 2. On main page,Admin taps the button called “Add a New Instructor” that is under the “Member” tab.   2.1.Admin taps the button called “Add a New Assistant” that is under the “Add A New Member” tab.  2.2.Admin taps the button called “Add a New Admin” that is under the “Add A New Member” tab.   1. The Admin fills up the information of Instructor - ( The Informations are name, surname , email address,and her/his department name ) - submits the form with tapping on confirmation button.   3.1. The Admin fills up the information of Assistant and submits the form with tapping on confirmation button.  3.2. The Admin fills up the information of Admin and submits the form with tapping on confirmation button.   1. Then AW Sheet displays alert box and asks the *admin,* If s/he is sure to add that Instructor is selected. 2. The admin, confirms that “Alert-box” by tapping ‘✓’ icon on his request. 3. AW Sheet checks the information of instructor that satisfies its own constraints , then adds the Instructor/Assistant on database and sends an email that contains her/his username and password with respectively. 4. The *Admin,* gets notification that operation has done successfully. |
| *Entry condition* | User is logged in to system as Admin |
| *Exit condition* | Instructor/Assistant/Admin added to the Aw Sheet. |
| *Exceptional Cases :* | *[addingExistingUser]:*It would basically be threwen when the admin tries to add same user again to system.  [*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.  [TimeOut]:This appears when the user does not do anything within a time that has already setted before. |

8.

|  |  |
| --- | --- |
| *Use case name* | InstructorSelectCourse-Mertali |
| *Participating actor* | Initiated by Instructor Or Assistant  Related to Admin  Interacts with AW Sheet |
| *Flow of events* | 1. Instructor taps the ‘Course’ icon where is it on bottom of the screen .Middle of the screen there is a “Select Courses” button that is nothing but invoking functionality to select a course to add on her/his schedule and connects that course with her/him. 2. AW Sheet displays a form that contains attributes of the course. 3. The instructor selects the course by touching on field and enters the course details, including numbers of midterms & quizzes, grading policy, assistant of course and attendance requirements. 4. Then alert box appears and asks the *admin,* If he is sure to remove that Instructor is selected. 5. Burak,the admin, confirms that “Alert-box” by tapping ‘✓’ icon on his request. 6. Afterall, Burak, the Admin, gets notification that the instructor has been deleted from the database successfully. 7. Feedback has been shown to the Instructor.(i.e. Request has been sended ! ) |
| *Entry condition* | User is logged in to system as Instructor/Assistant. |
| *Exit condition* | Selected course is linked to the Instructor/Assistant. |
| *Exceptional Cases :* | *[addingExistingCourse]:*It would basically be threwen when the admin tries to add same course again to system.  [*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.  [TimeOut]:This appears when the user does not do anything within a time that has already setted before. |

9.

|  |  |
| --- | --- |
| *Use case name* | AddCourses -Mertali |
| *Participating actor*  *Instances* | Initiated by Student  Interacts with AW Sheet |
| *Flow of events* | 1. Student logs in the application called ‘AW SHEET’ by her own mobile Phone with using his own password.Then He touches the ‘Login’ button. 2. Student taps the ‘Course’ icon where is it on bottom of the screen .Middle of the screen there is a “Select Courses” button that is nothing but invoking functionality to select a course to add on her schedule and connects that course with him.   2.1.Student types the name of the course in search box that is top of the screen and types the course name and selects the course that s/he is taking at the that semester.   1. Student taps `Add Selected Courses` to invoke add that course to his schedule. 2. Then AW Sheet displays an alert box and asks the *Student,* If s/he is sure to edit that course. 3. Student confirms that “Alert-box” by tapping ‘✓’ icon on his request. 4. Afterall, the Student, gets notification that selected course added. |
| *Entry condition* | User is logged in to system as Student |
| *Exit condition* | Selected course is added to Student Schedule |
| *Exceptional Cases* | [*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.  [TimeOut]:This appears when the user does not do anything within a time that has already setted before.  [addedSameCourse]:It would basically be threwen when the Student tries to add same course again to his schedule. |

10.

|  |  |
| --- | --- |
| *Use case name* | Take Attendance -Berk |
| *Participating actor* | Initiated by Instructor  OR Asistance,  Participated by Student |
| *Flow of events* | 1. Instructor launches the app. 2. Instructor opens attendance track tab. 3. Instructor selects track method from three option.   3.1. Instructor selects swipe method option.  3.2. Instructor selects qr method option.  3.3. Instructor selects list method option.  4.  4.1. Instructor reads the name of the student.If student is present swipes left.If the student not present swipes right[Check Mistakes].Tap the finish button at the end.  4.2. Instructor’s app opens an qr code that can be readable by students apps.When Instructor clicks on finish button, see the list of all student that read the qr code.[TimeOut]  4.3. Instructor see the list of student and mark the present ones and press the finish button. |
| *Entry condition* | * Instructor launches the app. * Instructor logged in successfully. * Students logged in successfully. |
| *Exit condition* | * Tapping on finish button. |

*Exceptional Cases*

*[Check Mistakes] : When Instructor tap top of screen and scroll it down, instructor see passed students when instructor stop scrolling the system returns to that student.Instructor can change the absence status of the student.*

[*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.

[TimeOut]:This appears when the user does not do anything within a time that has already setted before.

11.

|  |  |
| --- | --- |
| *Use case name* | EnterGrade -gülden |
| *Participating actor* | Initiated by Instructor OR  Assistant  Interacts with AW Sheet |
| *Flow of events* | 1. Instructor taps Courses icon from home page. 2. AW Sheet displays the courses that the Instructor is giving this semester. 3. Instructor selects a course by tapping. 4. AW Sheet displays exams which are registered before. 5. Instructor taps to a exam, then taps “Enter Grade for Exam” button. 6. AW Sheet lists the students of the course on screen. 7. Instructor enters the grades one by one for each student by typing. 8. After finishing, Instructor taps to save button. 9. AW Sheet saves the exam grades to database, then shows a “Saved successfully.” feedback to Instructor. |
| *Entry condition* | User is logged in to system as Instructor OR Assistant |
| *Exit condition* | Grade changes are successfully saved to database. |
| *Exceptional Cases* | **[TimeOut]:** This appears when the user does not do anything within a time that has already setted before.  **[*LostConnection*]:** This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System. |

12.

|  |  |
| --- | --- |
| *Use case name* | WithdrawCourse -Gülden |
| *Participating actor* | Initiated by Student  Interacts with AW Sheet |
| *Flow of events* | 1. Student taps the “*Courses*” Tab. 2. AW Sheet displays the courses student is taking. 3. Student taps to the course the student wants to withdraw. 4. Student taps to “more” icon of the course. 5. AW Sheet displays a drop-down menu including: withdraw course, check attendance, calculate attendance, check grade and calculate grade. 6. Student taps the “Withdraw Course” button on drop-down list. 7. AWSheet displays an alert box asking if the student is sure about withdrawing the course. 8. Student confirms the alert box. 9. AW Sheet sends a feedback as textbox saying “You have successfully withdrawn from the course.” |
| *Entry condition* | Student taking the course, logged into system |
| *Exit condition* | Student successfully withdraws the course. |
| *Exceptional Cases* | **[*LostConnection*]:** This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System. |

13.

|  |  |
| --- | --- |
| *Use case name* | Check Grade -Berk |
| *Participating actor* | Initiated by Student |
| *Flow of events* | 1. Student launches the app. 2. Student opens courses tab. 3. Student taps on the course name. 4. Student checks the grades. |
| *Entry condition* | * Student logged in successfully. * Student launches the app. |
| *Exit condition* | * Student check his/her grades. |

14.

|  |  |
| --- | --- |
| *Use case name* | SimulateGrade -gülden |
| *Participating actor* | Initiated by Student  Interacts with AW Sheet |
| *Flow of events* | 1. The student taps the “Courses” button. 2. AW Sheet lists the courses on screen. 3. Student taps the “more (. . .)” icon for a course 4. AW Sheet system displays a drop down menu. 5. The student taps the “Calculate Grade” button. 6. AW Sheet opens a new tab with some text boxes. Which are Midterms, Homeworks, Quizzes, Attendance points. 7. Student enters his predicted grades to given text boxes.***[TimeOut]*** 8. AW Sheet system calculates the predicted final grade depending on grading policy registered before.AW Sheet displays result on screen.***[MissingGrades] [LostConnection]*** |
| *Entry condition* | Student is logged into system, taking a course. |
| *Exit condition* | AW Sheet system calculates the predicted final grade. |

**[MissingGrades]:** To calculate the predicted final grade, all exam and assignment grades must be entered. Student should fill all blanks.

**[*LostConnection*]:** This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.

**[TimeOut]:**This appears when the user does not do anything within a time that has already setted before.

15.

|  |  |
| --- | --- |
| *Use case name* | CheckAttendancePercentage -gülden |
| *Participating actor* | Initiated by Student  Interacts with AW Sheet |
| *Flow of events* | 1. The student taps the “Courses” button. 2. AW Sheet lists the courses on screen. 3. Student taps the “more (. . .)” icon for a course 4. AW Sheet system displays a drop down menu. 5. The student taps the “Check Attendance” button. 6. AW Sheet turns records of student from selected course into a percentage value and shows to student in a pop-up tab. |
| *Entry condition* | Student is logged into a system and taking a course. |
| *Exit condition* | AWSheet displays the attendance percentage. |
| *Exceptional Cases* | **[*LostConnection*]:** This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System. |

*16.*

|  |  |
| --- | --- |
| *Use Case Name* | EditCourse -Mertali&Gülden |
| *Participating actor*  *Instances* | Initiated by Instructor  Interacts with AW Sheet |
| *Flow of events* | 1. Instructor taps the ‘Course’ icon where is it on bottom of the screen. 2. AWSheet displays the courses of the instructor. 3. The Instructor, selects the course on list. 4. AWSheet displays “course detail” page. 5. Instructor touches the option icon that is on right-top corner of the screen that opens a Drop-Down List. 6. Instructor, taps “Edit Course” on Drop-Down List. 7. AWSheet displays a Course Details Form. 8. The instructor selects the item that s/he wants to change (Allowed items are , Course syllabus , # of quizzes and midterms e.t.c) the course that s/he is giving. 9. AWSheet creates a text box for the field which is to be changed. 10. The instructor, changes the field , that s/he selected and taps Submit. 11. AWSheet displays an alert box and asks the *Instructor,* If he is sure to edit that course.. 12. Instructor, confirms that “Alert-box” by tapping ‘✓’ icon on his request. 13. Afterall, the Instructor, gets notification from AWSheet that the instructor has been edited in the database successfully. |
| *Entry condition* | User is logged in to system as Instructor/Assistant |
| *Exit condition* | Selected course is edited by Instructor/Assistant. |
| *Exceptional Cases* | [*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.  [TimeOut]:This appears when the user does not do anything within a time that has already setted before. |

*17.*

|  |  |
| --- | --- |
| *Use Case Name* | SelectCourse -Mertali |
| *Participating actor*  *Instances* | Initiated by Instructor  Interacts with AW Sheet |
| *Flow of events* | 1. Instructor taps the ‘Course’ icon which is at the bottom of the screen .Middle of the screen there is a “Select Courses” button that is nothing but invoking functionality to select a course to add on her schedule and connects that course with him. 2. AW Sheet creates a form that contains attributes of the course. 3. Instructor selects the course by touching on field and enters the course details, including numbers of midterms & quizzes, grading policy, assistant of course and attendance requirements. 4. AW Sheet displays an alert box and asks the *instructor,* If the instructor is sure to add that course to his/her schedule.. 5. The instructor confirms that “Alert-box” by tapping ‘✓’ icon on his request. 6. Afterall, the instructor gets a notification that the course has been added to his/her schedule successfully . |
| *Entry condition* | User is logged in to system as Instructor |
| *Exit condition* | Selected course is taken by Instructor |
| *Exceptional Cases* | [*lostConnection*]: This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System.  [TimeOut]:This appears when the user does not do anything within a time that has already setted before. |

*18.*

|  |  |
| --- | --- |
| *Use case name* | ForgotPassword -Korhan |
| *Participating actor*  *Instances* | *Initiated by Student*  *OR Instructor*  *OR Assistant*  *Interacts with AWSHEET* |
| *Flow of events* | 1. Student launches the app “AWSHEET” from his/her mobile device.   1.1. Instructor launches the app “AWSHEET” from his/her mobile device.  1.2. Assistant launches the app “AWSHEET” from his/her mobile device.   1. Student taps “Forgot Password” button from the login page. 2. Student fills the email of his/her existing account and taps the “Confirm” button. **[Invalid Email]** 3. A text box appears and notifies the Student if the email address exists in database a new password will be sent to that same email address. 4. AW SHEET generates a random password and changes the password for the Student. Then sends the new password to Student email. 5. Student receives the new password and successfully logs in with the new password. |
| *Entry condition* | * User downloads or already has the app “AWSHEET” |
| *Exit condition* | * User successfully logs in with new password. OR * Invalid email address is given and function cannot continue and needs to be repeated. |
| *Exceptional Cases* | **[*Invalid Email*] :** The email address filled by user must belong to an existing account from database. If not nothing will be sent to any email address. |

19.

|  |  |
| --- | --- |
| *Use case name* | GradeRequest -Korhan |
| *Participating actor*  *Instances* | *Initiated and confirmed by Instructor*  *Sent by Assistant*  *Interacts with AWSHEET* |
| *Flow of events* | 1. Instructor taps to “Courses” tab. Then taps on one of the courses. 2. Instructor taps “Grade Request” and fills the information about this request. Needed informations are; Course Name, Course Code and Quiz Number. 3. Assistant receives a notification from “AWSHEET” that contains the information filled by Instructor. 4. Assistant taps on the “Send Grades” button from the notification. And uses list method to fill quiz grades.**[Time Out]** 5. After filling the grades Assistant taps on the “Confirm” button. 6. Instructor gets a notification from “AWSHEET” that informs him/her the grades are given by the Assistant.   6.1 Instructor taps on the “Publish” button to share grades with students.  6.2 Instructor taps on the “Reject” button and send an email to Assistant explaining why he/she rejected it and request it again. |
| *Entry condition* | Instructor AND Assistant successfully logs in to AWSHEET. |
| *Exit condition* | Instructor publishes the grades.  OR  Instructor rejects and re-requests. |
| *Exceptional Cases* | [Time Out]: If Assistant stays idle for a determined time system automatically closes the function tab. |

20.

|  |  |
| --- | --- |
| *Use case name* | GetAttendance -Korhan |
| *Participating actor*  *Instances* | *Initiated by Instructor*  *Interacts with AWSHEET* |
| *Flow of events* | 1. Instructor taps to “Courses” tab. Then taps on one of the courses. 2. Instructor taps “Attendance Request” for Ps and Lab courses. 3. “AWSHEET” sends attendance of selected Ps and Lab courses to Instructor. And merge them with course attendances. **[Time Out]**   3.1 Instructor taps on the “Publish” button to share the attendance with students.  3.2 Instructor taps on the “Wait Until a Date” button and chooses a date to publish the attendance. |
| *Entry condition* | * Instructor successfully logs in to AWSHEET. |
| *Exit condition* | * Instructor publishes the attendance.   OR   * Instructor selects a date to publish the attendance. |
| *Exceptional Cases* | **[Time Out] :** If Instructor stays idle for a determined time function stops running. |

21.

|  |  |
| --- | --- |
| *Use case name* | ViewSchedule -Korhan |
| *Participating actor*  *Instances* | *Initiated by Student*  *OR Instructor*  *OR Assistant*  *Interacts with AWSHEET* |
| *Flow of events* | 1. Student taps “Schedule” button from home page.   1.1 Instructor taps “Schedule” button from home page.  1.1.1 Schedule of the Instructor is displayed with the informations including; Course Name, Course Room and Course Hour.  1.2 Assistant taps “Schedule button from home page.  1.2.1 Schedule of the Assistant is displayed with the informations including; Course Name, Course Room and Course Hour.   1. Schedule of the Student is displayed with the informations including; Course Name, Course Room and Course Hour. |
| *Entry condition* | User is successfully logged in to the app. |
| *Exit condition* | User views the calendar. |

22.

|  |  |
| --- | --- |
| *Use case name* | SendAnnouncement -Berk |
| *Participating actor*  *Instances* | *Initiated by Instructor OR*  *Assistance* |
| *Flow of events* | 1. Instructor opens tab and selects course. 2. Tabs on new announcement button. 3. Sends the announcement. |
| *Entry condition* | User is successfully logged in to the app. |
| *Exit condition* | Instructor sends announcement. |

23.

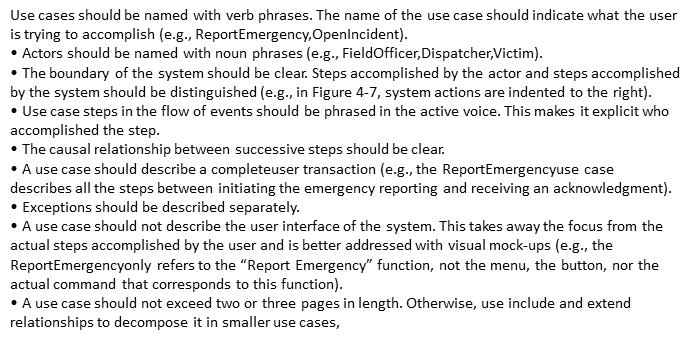
|  |  |
| --- | --- |
| *Use case name* | ViewNotification -Gülden |
| *Participating actor*  *Instances* | *Initiated by Student*  *OR Instructor*  *OR Assistant*  *Interacts with AWSHEET* |
| *Flow of events* | 1. Student notices the notification.    1. Student notices by seeing the notification on notification bar.    2. Student notices the notification while using the app, by seeing the notification icon. 2. Student taps to the notification. 3. AW Sheet displays the details of the notification. 4. Student views the notification. |
| *Entry condition* | Student gets a new notification. |
| *Exit condition* | Student views the notification. |
| *Exceptional Cases* | **[*LostConnection*]:** This exception will be throw when the Instructor or Admin taps a action button.That the instructors are not able to go to the AW Sheet System. |

***-USE CASES***

***COLOR:MEANS DONE AND MEANS DONE & CHECKED*** +lar scenariolarda ki eventler \*lar diagramları yapıldı demek. #sequentialı yapıldı***(Birleştirilecek)***

1. Register -korhan +\*#
2. Login -Berk + \* #
3. Logout -Berk + \* #
4. Reset Password -Berk+ \* #
5. ///{MERGED-Duplicates}
6. Delete Instructor(Case -1 : Admin Case-2:Student)-Mertali+\*#
7. Add User-Mertali+\*#
8. ///{MERGED-Duplicates}
9. Add Courses Student-Mertali+\*#
10. Take Attendance(Case-1:QR Case 2:Swipe EXCEPTIONAL CASE:) -Berk+ \*
11. EnterGrade(Actor: Instructor OR Assistant)-gulden+\*#
12. Withdraw Course -gulden+\*#
13. Check Grade(Actor: Student) -Berk+ \* #
14. Simulate grade(Actor: Student)-gulden+\*#
15. Check Attendance percentage(Actor: Student)-gulden+\*#
16. Instructor edits course-Mertali&Gülden+\*#
17. Select Course-Mertali+\*#
18. UserForgetsPassword-korhan+\*#
19. Assistant sends quiz grades -korhan+\*#
20. Assistant sends attendance -korhan\*#
21. ViewSchedule -korhan\*#
22. Sending Notification - Instructor/Assistant -Berk + \* #
23. Viewing Notification - Student -gulden\*#

***BASLAMADAN ONCE ..!***



***Mantıklı!***

***-SCENARIOUS***

***COLOR:MEANS DONE AND MEANS DONE & CHECKED AND MEANS NOT FINISHED AND + MEANS ORDERED AS LIKE LISTED.***

***\***

1. Instructor | Selects Course -Mertali +
2. Instructor/Assistant Registration-Gulden&Mertali +
3. Student Withdraws Course.-Mertali+
4. Login Operation-Mertali +
5. Student Course Addition -Mertali +
6. Instructor edits course attributes(assistant, syllabus,# of quizzes,e.t.c.)-Mertali +
7. Admin Adds Course List -Mertali+
8. Admin determines Add/Drop-Mertali+
9. Instructor | Takes attendances himself/herself (by swipe function)-gülden+
10. Instructor | Takes attendances himself/herself (by QR CODE)-gülden+
11. Instructor | Takes attendances himself/herself (by Table)-gülden
12. Student Scans QR Code-gülden+
13. Assistance & Instructor | Enters Grade -gülden&berk+
14. Student Registration -korhan +
15. Student Tracks Grade -korhan +
16. Student Tracks Attendance -korhan +
17. Student Receives Announcement -korhan +
18. Student calculates his/her grade - korhan+
19. Reset Password. -Berk+
20. Forget Password. -Berk+
21. Assistant sends quiz grades to instructor -berk +
22. Assistant sends attendance to instructor -berk +
23. Student checks his/her schedule. -Korhan+
24. Admin deletes instructor-Gulden&Mertali

{｡^◕‿◕^｡} {｡^◕‿◕^｡} {｡^◕‿◕^｡} {｡^◕‿◕^｡} {｡^◕‿◕^｡} {｡^◕‿◕^｡}

MOCK-UP BASLANDI..!

--------------------------------------------Challenge accepted..!---------------------------------------

🌔 ☁️ ☄️ ☁️ 🛸

|  |
| --- |
| Assistant’ın mock-upları kaldı ama sorun olacağını zannetmiyorum ilerleyen zamanlarda updateleriz & önceliğimiz(Puan alacağımız yer RAD oldugundan öncelik diagramlar ama bitti sayılır … Table haline getirip buraya koyacağım…) |

🌃 🦌