**HOME ASSIGNMENT SERVICE**

**Introduction:**

Home Assignment App is a web based application that gives a common platform for Faculty and Student to upload and download Assignments. This provides some basic services to Authenticate an user and provide some functionalities he/she can perform like upload/download assignments, Notifications when some events are triggered (Assignment created, downloaded, inprogess etc)

**Pre-Requisites:**

1. Platform is divided into classes and each classes a set of teachers assigned.
2. Each class is mapped to a unique faculty (this is for piloting the project and features can be added later)
3. Each class has set of students assigned and for piloting student is currently tagged to one class
4. For this Application, Assignment of class and students is done already and data been updated in database.
5. An admin application is intended to be developed to perform some admin level activities like, creating a class, assigning faculty and students.
6. User Registration is out of scope and data added manually.

**Technical Design:**

**Class Definitions:**

User: Detailed information for an user

UserRole: Roles Defined. Each User will be assigned a Role.

Class: Defines Class definition

ClassFaculty: Defines Faculty assigned to a particular Class

ClassStudent: Defines Students assigned to a particular Class

Assignment: An assignment record, associated to Class, Faculty who created the assignment, and assignment details also status (refer status class)

AssigntSubmission: Assignment record, with mapped student contains submitted assignment details

AssignmentStatus: This are different stages of Assignment status. Created, Downloaded, In Progess, Submitted, Verfied, Completed.

**Use Cases:**

1. Authenticate User:
   1. Factulty/Student Access the web client. Login screen is provided.
   2. He/She provide credentials for login. (Registering for the web application is out of scope)
   3. Request to Identity server to authenticate user. Using Hybrid Authentication (Login request along with key)
   4. If credentials are correct, response with access token and claims (contains roles access)
   5. If credentials are incorrect, response 404 Unauthorized.
2. Faculty Accessing Application
   1. After authorizing, Faculty user as authorized.
   2. Client application request API call, GET Class to Web App
   3. Web App Validates the request and checks Role Permission to see a faculty role is accessing the API. Also does Key Chain validation to check if Faculty is part of the current Class.
   4. If validated, then call Business Service to get Class Response which contains Assignment Info, Student Info etc.
   5. If user is not Authorized (not a faculty role or key chain validation fails, return 404 Unauthorized)
   6. Faculty access Home Page, where current Assignments displayed. Has option to Create Assignment.
3. Student Accessing Application
   1. Student is Authenticated.
   2. Client application request API call, GET Class to Web App
   3. Web App Validates the request and checks Role Permission to see a Student role is accessing the API. Also does Key Chain validation to check if Faculty is part of the current Class.
   4. If validated, then call Business Service to get Class Response which contains Assignments Info
   5. If user is not Authorized (not a faculty role or key chain validation fails, return 404 Unauthorized)
   6. Faculty access Home Page, where current Assignments displayed. Has option to Create Assignment.
4. Faculty Creates Assignment:
   1. Check faculty user is still authenticated, if token expired, redirect to Login screen.
   2. Request to create assignment.
   3. Client app does an empty Post to web app to get required Assignment Model definition back
   4. Faculty enter required Assignment info and uploads the assignment using File Upload.
   5. When form posted, Client application does basic validation and async call post to Web API
   6. Web App does check for permissions and calls Business service which creates a database record which has a unique GUID for file. Web App then call File System Service which creates a record in MSMQ.
   7. As request is sent to File system service, sends success message to Faculty. ‘Upload process started’
   8. File upload is done using Message Queue, and once upload is done, sends a call back to Business Service to notify upload is finished.
   9. Business Service will update database to change status for Assignment to ‘Uploaded’ and sends a request to Notification Service which will send notifications (Email or Text, and additional notification can be updated) to both Faculty and Student.
5. Student Access Assignment:
   1. Student receives a notification. ‘Assignment has been added’
   2. Student access Assignment. Client Application will authenticate
   3. If logged in, Client App send an async call Get Request to Web API
   4. Web APP does check for permissions and calls Business Service to get the Assignment details.
   5. Business service access the file info, and not the data for faster response.
   6. Sends Assignment info to Web App and then Client Application.
   7. Student given options to Download, which then again sends an async call Get to download the resource.
   8. Web app does check for permissions and class Business Servie to get File from File System Service.
   9. File system service download data and send stream back.
   10. Client App, created a link and download option is intiated.
   11. Business Service will update database for Assignment Status to ‘Downloaded’
   12. Student after viewing downloaded assignment will initiate, Start to notify he started assignment.
   13. Client app does a post call to Web API to update status of Assignment to ‘In Progress’
6. Student Submit Completed Assignment:
   1. Student logins, and since status is ‘In progress’ he is provide with option to upload the completed assignment.
   2. When requests to upload assignment, Client app does an async call post to upload assignment.
   3. Web Api checks for permissions and call Business Service for file uploading
   4. Process is like upload file.
   5. Once upload is completed, Business service send request to Notification service to send notifications to Student and Faculty.
   6. Updates Assignment Status to ‘Submitted’.
7. Faculty review Submitted Assignment:
   1. Faculty receives notification assignment submitted.
   2. Logins to Client Application to review.
   3. Get Assignment will show status and also link to download file. Similar to Student access assignment.
   4. Once download is completed, checked and verified clicks Verified.
   5. Client Application will send a post call to update status to ‘Verified’
   6. Business service will send request to Notification Service to send notifications to Student.

**Web Api:**

/api/faculty/{id}/Class/{classid}/Assignments

/api/faculty/{id}/Class/{classid}/Assignment/{assignmentid} Get

/api/faculty/{id}/Class/{classid}/Assignment/{assignmentid} post

/api/faculty/{id}/Class/{classid}/Assignment/{assignmentid}/Download/{fileId} post

/api/faculty/{id}/Class/{classid}/Assignment/{assignmentid}/Verify Post

/api/Student/{id}/Class/{classid}/Assignment/{assignmentid} Get

/api/Student/{id}/Class/{classid}/Assignment/{assignmentid}/Download/{fileId} Get

/api/Student/{id}/Class/{classid}/Assignment/{assignmentid}/Start POST

/api/Student/{id}/Class/{classid}/Assignment/{assignmentid}/Submit POST

**Notes:**

Key chain validation is done using user id and Class part of the uri to determine if user is part of the class.