PROJECT DELIVERABLE 2
USE Case Contracts (40%)

USE CASE :Sign up & login to the application

OC 1:

Operation: ChooseRole()
Cross References: UseCase :Sign up & login to the application
PreConditions: The user must be an eligible person having complete knowledge of his role.
PostCondition:

- An instance “role” of UserRole is created.
- The role is identified as either tourist or service provider.
- The instance role’s string attribute has been assigned a value.

OC 2:

Operation: signUp()
Cross References: UseCase :Sign up & login to the application
PreConditions: The user must be an eligible person having complete knowledge of his role.User must agree to provide all required information and sign contract in case he is a service provider.
PostConditions:

- User sign up successfully as a tourist or service provider and be able to complete his profile.
- Instance of Email class “email” is created. This instance is used in communication with the user.
- The user has assigned a unique username and password.
- Password and username attributes of Signup class are assigned a value.

OC 3:

Operation: login()
Cross References: UseCase :Sign up & login to the application
PreConditions: User must have an account.
PostConditions:

- User logged in as tourist or service provider.

- If user is a service provider, he/she can offer his/her services as a list “Services” of Service class instances is created.
- If user is a tourist, he/she is able to avail services.
- If user is tourist, user has a TouristID.
- The service provider has an instance of Contract that he/she has signed with application.
- If user is a service provider, user has a ServiceProviderID.
- Tourist/service provider has an instance of NewsFeed. Through that instance user is able to delete, update, approve, display content such as videos, pictures and blogs.
- User is able to customize his/her user interface.

OC 4:

Operation:AuthenticateUser()

Cross References: UseCase :Sign up & login to the application

PreConditions: User has opened the application and wants to login/signup.

PostConditions:

- The user is proved to be a human and declared safe for further login/signup operation.
- The user is proved to be the genuine owner of the account.
- It is declared that the user is not a scam.

OC 5:

Operation:completeProfile()

Cross References: UseCase :Sign up & login to the application

PreConditions: User has successfully signed up in the application.

PostConditions:

- The user’s all the required information is available in his/her profile.

USE CASE : Manage pictures, videos and blogposts

OC 1:

Operation:AddPicture(picture:Picture)

Cross References: UseCase : Manage pictures, videos and blog posts

PreConditions : User has successfully logged in into the application.

PostConditions:

- An instance of Picture class is added into List of Picture objects in NewsFeed class.
- The instance will have Picture name, DateTime and a unique imgId.
- The picture is available in the NewsFeed with other pictures.

OC 2:

Operation:AddVideo(video:Videos)

Cross References: UseCase : Manage pictures, videos and blog posts

PreConditions : User has successfully logged in into the application.

PostConditions:

- An instance of Video class is added into List of Video objects in NewsFeed class.
- The instance will have Video name, DateTime and a unique vidId.
- The video is available in the NewsFeed with other videos.

OC 3:

Operation:AddBlogPost(blogSpot:BlogSpots)
Cross References: UseCase : Manage pictures, videos and blog posts
PreConditions : User has successfully logged in into the application.
PostConditions:

- An instance of BlogSpots class is added into List of BlogSpots objects in NewsFeed class.
- The instance will have BlogSpots name, DateTime and a unique BlogPostid.
- The BlogSpots is available in the NewsFeed with other BlogSpots.

OC 4:

Operation:DeletePicture(picture:Picture)
Cross References: UseCase : Manage pictures, videos and blog posts
PreConditions : User has successfully logged in into the application and is the owner of the picture he/she wants to delete.
PostConditions:

- An instance Pictures class is deleted from List of Pictures objects in NewsFeed class.
- The Picture is made unavailable from the NewsFeeds.

OC 5:

Operation:DeleteVideo(video:Videos)
Cross References: UseCase : Manage pictures, videos and blog posts
PreConditions : User has successfully logged in into the application and is the owner of the video he/she wants to delete.
PostConditions:

- An instance Videos class is deleted from List of Videos objects in NewsFeed class.
- The Videos is made unavailable from the NewsFeeds.

OC 6:

Operation: DeleteBlogPost(blogSpot:BlogSpots)
Cross References: UseCase : Manage pictures, videos and blog posts
PreConditions : User has successfully logged in into the application and is the owner of the BlogSpots he/she wants to delete.

PostConditions:

- An instance BlogSpots class is deleted from List of BlogSpots objects in NewsFeed class.
- The BlogSpots is made unavailable from the NewsFeeds.

OC 7:

Operation:DisplayContentRandomly(PicturesList : List,VidesoList List, BlogSpots : List) {static}
Cross References: UseCase : Manage pictures, videos and blog posts
PreConditions : User has successfully logged in into the application.
PostConditions:

- The Newsfeed is displayed where content such as videos, pictures and blogSpots are displayed randomly(No associations needed as the function is a static one).

USE CASE : Manage Services

OC 1:

Operation: AddService(SPid:string,TSid:string,details:ServiceDetails,discounts:Discounts)
Cross References: UseCase : Manage Services
PreConditions : Service provider has successfully logged in into the application.
PostConditions:

- For a particular tourism spot, the user has added his services into the services list at that spot.
- Through this operation, details of services provided by user are added and details of discount such as discount name, discount type, discount price, discount validity are added. These are basically the attribute of the instance of Discount class.
- All the details about a particular service at a particular tourism spot are available after this operation.

OC 2:

Operation: DeleteService(serviceID:string)
Cross References: UseCase : Manage Services
PreConditions : Service provider has successfully logged into the application. The service provider is the owner of the service he/she wants to delete.
PostConditions:

- For a particular tourism spot, user has deleted his services instance from the List of services at that spot (services:List<Serivces> of the tourism spot class).
- The service is no longer available in the UI hierarchy.

OC 3:

Operation: UpdateService(serviceID:string)
Cross References: UseCase : Manage Services
PreConditions : Service provider has successfully logged into the application. The service provider is the owner of the service he/she wants to update.
PostConditions:

- The details about the service and discount details are updated.
- The services:List in TourisumSpots class will have updated instance of Service.

USE CASE : Manage and facilitate Searches

OC 1:

Operation: RenderHirarchy(){static}
Cross References: UseCase : Manage and facilitate Searches
PreConditions : The user must have logged in successfully.
PostConditions:

- The hierarchy of all the services provided according to the provinces, cities and spots is displayed.(No associations needed as the function is a static one).

USE CASE : Manage content monitoring and control

OC 1:

Operation: RequestUpload(string)
Cross References: UseCase : Manage content monitoring and control
PreConditions : The user must have logged in successfully. The user wants to upload some content.
PostConditions:

- The request to upload some content is first viewed and judged by admins and moderators.
- Content can be Pictures,Videos and BlogSpots. So these classes have an association with Admins&Moderators class.
- If the content is irrelevant, the request is denied.
- If the content is inappropriate, the request is denied.
- If the content source is not mentioned, the request is rejected.
- If content is relevant, appropriate and the source is mentioned then it is allowed to be published.

OC 2:

Operation: RequestModification(string);
Cross References: UseCase : Manage content monitoring and control
PreConditions : The user must have logged in successfully. The user wants to modify some content he/she owns.
PostConditions:

- The request to modify some content is first viewed and judged by admins and moderators.
- Content can be Pictures,Videos and BlogSpots. So these classes have an association with Admins&Moderators class.
- If the content is irrelevant, the request is denied.
- If the content is inappropriate, the request is denied.
- If the content source is not mentioned, the request is rejected.
- If content is relevant, appropriate and the source is mentioned then it is allowed to be published.

OC 3:

Operation: ApproveUpload(string);
Cross References: UseCase : Manage content monitoring and control
PreConditions : The user must have logged in successfully. The user wants to upload some content. The content is appropriate, relevant and source is mentioned.
PostConditions:

- The content is added into the PicturesList : List OR VidesoList : List OR BlogSpots : List in the NewsFeed class depending upon the type of content(if it is picture, videos or blogspot).

OC 4:

Operation: ApproveModification(string);
Cross References: UseCase : Manage content monitoring and control
PreConditions : The user must have logged in successfully. The user wants to modify some content. The modified content is appropriate, relevant and source is mentioned.
PostConditions:

- The modified content is added into the PicturesList : List OR VidesoList : List OR BlogSpots : List in the NewsFeed class depending upon the type of content(if it is picture, videos or blogspot).