| **Feature** | **MVC** | **Class** | **Attributes/Variable** | **Methods** |
| --- | --- | --- | --- | --- |
| User Authentication  (Registration & Login) | Model | UserAuthentication | * List<User> users | * User register(String username, String email, String password) * User login(String username, String password) * logout(User user) |
| View | AuthView |  | * displayLoginScreen() * displayRegistrationScreen() * showSuccessMessage(String message) * showErrorMessage(String error) |
| Controller | AuthController | * UserAuthentication auth | * handleRegistration(String username, String email, String password) * handleLogin (String username, String password) * handleLogout(User user) |
| Homepage | Model | User  *Stores user details, reading goals, and channels* | * String username * String email * String subscriptionStatus * String paymentType * int paymentInfo * List<ReadingGoal> readingGoals * List<Channel> channels * Library library * String userSubscription * ImageIcon profile | * updateProfile (String newEmail, ImageIcon newPic) |
| Reading Goal  *Represents the user’s defined reading targets*  *(# of books per month/year)* | * Int goalID * Int targetBooks * String timeframe (monthly, yearly,etc.) * Int booksRead |  |
| Library  *Manages the books the user has read and track progress* | * List<Book> booksRead * List<Book> booksToRead |  |
| View | HomepageView |  | * displayReadingGoals(List<ReadingGoal> goals) * displayLibrary(Library library) * displayChannels(List<Channel> channels) |
| Controller | HomepageController | * User user | * getReadingGoals() * getLibrary() * getChannels() * updateReadingGoal(ReadingGoal goal) |
| Track Progress | Model | Book  *Represents a individual book* | * Int bookID * String name * String author * Boolean isRead * double progress (0 to 100) |  |
| ReadingProgress  *Tracks progress for books read and books user want to read* | * List<Book> booksRead * List<Book> booksToRead | * addBookToRead(Book book) * markBookAsRead(Book book) * updateProgress(Book book, double progress) |
| View | BookProgressView |  | * displayBooksRead(List<Book> books) * displayBooksToRead (List<Book> books) * displayReadingProgress (Book book) |
| Controller | BookProgressController | * ReadingProgress readingProgress | * addBookToRead(String name, String author) * markBookAsRead(String name) * updateProgress(String name, double progress) |
| Subscription Management | Model | PaymentSub | PaymentType paymentType  int paymentNumberInfo  String paymentFName  String payment LName | PaymentType paymentType = [credit, bank paypal] |
| SubPlans | SubPlans sub  String subStatus  String subDescription  SubPlans currentSub | SubPlans sub = [basic, premium, vip] |
| View | ViewPayment | PaymentType paymentType  int paymentInfo  String paymentFName  String payment LName | displayPaymentType (String paymentType)  displayPaymentInfo (int paymentInfo)  displayPaymentName (String paymentFName, String payment LName) |
| ViewSubPlans | SubPlans sub  boolean subStatus  String subDescription  SubPlans currentSub | displaySubPlans (SubPlans subPlans)  displaySubDescription (String subDescription)  displayCurrentSub (String currentSub) |
|  | AddPaymentInfo | String paymentFName  String paymentLName  String paymentType  Int paymentNumberInfo | addPaymentFullName (String paymentFName, String paymentLName)  addPaymentType (String paymentType)  addPaymentInfo (int paymentNumberInfo) |
| Controller | ModifySubs | boolean subStatus  SubPlans sub  SubPlans currentSub | addSub (SubPlans sub, boolean subStatus)  deleteSub (SubPlans currentSub, boolean subStatus)  changeSub (SubPlans sub) |
| Channel Management | Model | Discussion Channel Model | * (String)Username (ID) * (ArrayList<String>)Subscribed Channels * (Boolean)Subscription Status |  |
| Channel Object | * (String)Channel name * (String)Channel Owner * (Int) Number of Subscribed users * (Int) Number of viewers on a discussion channel |  |
| Comment Object | * (String) UserName * (String) Comment | * (String)Get UserName * Set Comment (<String> comment) |
| View | Channel View | * Image Book * Comment Comments * (Button) Addcomment * (Button) Like Comment | * (Image) GetBook * (Image) SetBook |
| Channel  Container | * (JPanel) Channel Container * (List) Channel * (JButton) Create Channel |  |
| Controller | Channel Controller |  | * subscribeToChannel() * unsubscribeToChannel() * likeComment() * unlikeComment() * replyToComment() * deleteReplyToComment() * reportAComment() * createChannel() * deleteChannel() * addCollaborators() |
| MeetingManagement | Model | Meeting | * meetingDate (date) * meetingTime(time) * meetingName(string) * meetingLink(String) | * createMeeting() * deleteMeeting() * editMeeting() * findMeeting() |
| Model | MeetingsList | * meetingList (undecided data structure) - Meeting passed through as variable | * viewMeetingsDay() * viewMeetingsWeek() * addMeeting() |
| View | MeetingListView |  | * displayMeetingView() |
| View | MeetingsCalendarView |  | * displayCalendarView() |
| Controller | MeetingsListViewController | * meetingList |  |
|  | Controller | MeetingsCalendarController | * meetingsList |  |

| **Team Member ID + Name** | **Contribution to the Particular Assignment** |
| --- | --- |
| 943291592 + Britney Wang | I worked on both the API design and the code implementation for the Subscription Management MVC. |
| Stephanie Goodie-Obi | I worked on the Channel Management MVC, and all the classes contained in the code logic |
| Sereina Williams | I worked on both the API design and the code implementation for the Homepage and TrackProgress MVC. |