**3. You should have a folder named design containing**

**(a) High-level design document (in HTML).**

**i. Containing one or more UML class diagrams showing your packages and the**

**dependence relations between these packages. Also show library packages**

**or third-party packages that are used.**

**Diagram, schematic

Description automatically generated**

**ii. For each of your packages, a short description of the purpose of the package.**

The implementation of the project is done with the MVC software design pattern. Hence the main packages used are as follows:  
  
1) Model Package:

The model package is responsible to update the view package after it receives the manipulations from the controller package. The model package is the central component of this pattern. It is responsible for directly managing the data application data, logic, and rules of the application.

Contains: exit.java, fontSize.java, main.java, panelInfo.java etc.

2) View Package:

Any representation of the application and information is handled with the help of this package. Multiple different customizations according to our requirements are done so that we can create a GUI based on it which serves our purpose.

Contains: addpanel.java, ComponentMover.java, ComponentResizer.java, FileFilter.java, JpanelSlider.java etc.

3) Controller Package:

The purpose of the controller package is to accept the input and convert it to use for the model and the view packages. It performs interactions on the data model objects. It receives the input, validates it optionally and then proceeds to pass it to the model.

Contains: FontChooser.java, Jtext.java, MainFrame.java, etc.

4) Test Package:

The purpose of the test package is to test all the main classes in the different packages above. We have used Junit for the purpose of the testing and have successfully tested 7 different classes. This proves that they can work independently.

**iii. For any third-party packages, you use, explain the purpose of the package**

**and where its documentation can be found.**

We have created the application interface GUI using the JTattoo, which is used for developing GUI with the help of swing applications. It enhances the look in comparison to the standard JDK. The documentation can be found at the following web address along with installation and usage procedure:

http://www.jtattoo.net/index.html

**iv. For each of your packages, one or more class diagrams showing the classes**

**and interfaces inside the package, associations between them and associations with classes in other packages that are depended on.**

**Diagram, schematic

Description automatically generated**

**Diagram

Description automatically generated**

**Diagram, schematic

Description automatically generated**

**v. For each class and interface, a description of the purpose of the class. I.e.,**

**what it represents or does.**

**FileFilter:** The main functionality of FileFilter is to open, save, save as and exit the file. This class is a filter that has used to returns abstract pathnames denoting the files and directories in a given directory so that it can be opened and saved properly.

**Shape:** This class is for managing all the shapes. It has functionalities to move the components on the slide anywhere in x-y direction.

**AddStates:** It will manage all the states.

**Transition:** Transition class is to maintain state’s transitions like next state, previous states.

**Camera:** This class is responsible to maintain camera position, zooming factors like zoom-in, zoom-out.

**MainFrame:** It is the GUI of whole app that’s shows buttons like- File, Home, View, Edit, Camera, Presentation, Transitions, Undo as well as attribute panel and panel (left slides panel and the main slide panel).

**StatesPanel:** This class is for adding, updating panels. It actually adds the slides by clicking the Add Slide button which has shown in Home Panel.

**FontChooser:** This class helps to select font for writing text on the slides.

**FontType:** To get the Font type and to use the font in the text box, this class has used.

**ColourChooser:** ColorChooser class is o choose the color for text, shapes.

**FontSize:** To change the font-size, color, and its font-family in our project by using FontSize class.

**Main:** It’s the main class from where our code is running.

**States:** This class controls state’s position, it’s height and width.

**PanelInfo**: It stores all the info about the panels we add into it.

**CameraController:** This class controls camera’s position as well as zoom-in, zoom-out functions.

**StateController:** StateController class is responsible for modifying states. It controls adding, deleting, updating of states.

**(b) A test plan that explains how you intend to test each package and class.**

-For testing purpose we use Junit, it helps for to writing and running tests. By using Junit in our preezoom presentation software. While testing the classes of our presentation software while testing the each package and classes, The program runs automatically and they check their own results and it generate the feedback automatically.

**Model**

It provides complete variable objects to the Controller layer, like data, tools, and static files so that you can have reliable and maintaipulate the data resource to test against. The data or information provided by the model package can range variables by testing environment and the ones generated by user customized tools .This layer contains only the functionalities to read and update test data and use tools.while testing its classes.

**Main:** The main class from where our code is running is running successfully and the feedback is generated.

**Panelinfo:** During testing the panel information are updated in the panel infobar.

**Font size:** The font size is working well the user can change the font size according to their needs .The color and text theme are working perfectly the user can use these options for their slide.The code for changing color,text,size are running successfully and the feedback is enerated.

**Camera controller**:The controller should control the camera movements ,The camera movements are still in the developing stage.

**State controller:**The state controller is still in developing stage.We are facing some error while updating and modifying the states testing it will be recovered soon and it will be implemented as soon as possible.

**Controller**

It is the interactions between your test cases and the Model layer. The controller layer is responsible only for providing functions or libraries for the overall test architecture. It gets data or information created by the tool or from the model layer, tests the given data, and returns the outcome for the test cases while testing phaseThe controller layer contrls all the logic involved in while interacting with test cases and will deliver data from the model to the testsuite layer to perform testing.

**FileFilter:** In file filter the open and exit operations are working but we are still implementing and working on with save and save as option.

**AddStates:** All states are manageable but still we are in the developing and testing stage.

**Camera:** The main functions are zoom in zoom out,The functions are coded properly but while testing there are some warnings but still the functions are in implementation level.

**Shape:** It has functionalities to move the components on the slide anywhere in x-y direction. The orientations are coded properly but we are still in the testing level we are facing some error while testing. The rectangle,line,circle,oval are implemented but we are facing issue while drawing on the silde it will be implemented as soon as possible.

**Transition:** The transitions of state like next state, previous states are still in the developing stage.

**View**

This layer handles user interaction. Its main job is to make sure the data is presented to the user in a readable and user friendly. The other function is to is submitting or generating the user actions to the Controller along with the user data that are provided. This layer handles the user interaction allows you to change the functions in preezoom presentation software according to the user needs .While testing this layer, some of the classes operation are tested and implemented and some of the class operations are still in the testing stage.

**StatesPanel:** The slide panel is working in a perfect manner. The test results are generated automatincally in Junit.

**FontChooser:** While testing the code runs properly the user can choose the font according to their needs and the feedback is generated and the result is obtained.

**FontType:** The font type option is working properly and it is implemented . The code is tested properly and it is runnable,the result is generated.

**ColourChooser:** The user can choose the color for text, shapes. like red, pink, blue, green etc., in their slide. The code is tested and it is implemented properly.

**MainFrame:** The GUI-Graphical User Interface of whole app that’s shows buttons like- File, Home, View, Edi options or functions are tested and it is implemented properly but Camera, Presentation, Transitions, State controller functions are still in the developing stage. It will be tested and implemented in the Iteration 2.

**(c) A short document that explains how to compile and execute your code.**

To compile and execute the code, we need proper platform to run the project.

-Java SE Development Kit :

In order to compile and execute our Java project “Prezoom, it need to install a latest version of software program called Java SE Development Kit (or JDK for short, and SE means Standard Edition).

-Java Runtime Environment :

A JDK contains JRE (Java Runtime Environment) basically which is the core of the Java platform that enables running Java programs on the computer. The JRE includes **JVM**(Java Virtual Machine) that runs Java programs by translating from bytecode to platform-dependent code and executes them (Java programs are compiled into an intermediate form called bytecode).

-Operating system :

For compiling and executing, operating system windows, Mac or Linax is suitable for our java project PreZoom.

-Text Editor :

To code for the program, a text editor eclipse has been used. Any text editor is fine such as netbeans, codeblocks.

-JTattoo:

To improve application with a user interface, Jtattoo has used which consists of several different Look for swing project. It is an essential third party package which is included with the project.

**(D) A short document that lists for each team member, which part they worked on.**

* Each Member of the group has contributed to the project in the following manner:
* Shujana Mostafa - Worked on User Stories, State Diagrams, Requirement Document, high level design document. Worked on Java files ComponentMover, ComponentResizer, Display, Filter, JPanelSlider, noShape.
* Christiani Monish Preethi - Worked on lexicon, Sequence Diagrams, Use cases, high level design document. Worked on Java files Circle, Line, Ellipse, Parallelogram, Rectangle, Square, Shape, Exit.
* Utkarsh Trivedi - Worked on wireframe diagram, Repository, List of requirements, Sequence diagrams, Class diagrams, PPT, high level design document. Worked on Java files Mainframe, Controller, StateManager, PopUp, FontSize, Presentation, Zoom.
* Each member has worked in multiple files for implementation due to MVC. The bug review and test cases in Junit were done by all the members together. Also various diagrams and documents were reviewed, edited and modified simultaneously.

**4. You should have a folder named src (or source) that contains your source code.**

**(a) Each class should be documented using JavaDoc (for Java), TSDoc (for Typescript), or Dox (for Haxe). Each public method should also be documented.**

|  |  |
| --- | --- |
| **Class Name** | **Functionality** |
| **Main** | It’s the main class from where our code is running. |
| **MainFrame** | It’s the GUI of whole app that’s shows buttons (File, Home, View, Edit, Camera, Presentation, Transitions), attribute panel and panel (left slides panel and the main slide panel). |
| **JPanelSlider** | It swaps the main slide to left and right using media buttons |
| **FileFilter** | Its functionality is to open, save , saves and exit the file. |
| **Addpanel** | It actually add the slides by clicking the Add Slide button(show in Home Panel). |
| **ComponentResizer** | The main functionality of this class is to resize the components which we add on the slides like shapes and text boxes. |
| **ComponentMover** | It’s functionality is to move the components on the slide anywhere in x-y direction. |
| **FontSize** | We change the font-size, color, and its font-family by using this class. |
| **PanelInfo** | It stores all the info about the panels we add into it. |
| **JText** | This class helps to add text on the slides. |