DESIGN JUSTIFICATION DOCUMENT

CS346: SOFTWARE ENGINEERING LABORATORY

Group 11
Project 7: Paint Application

Samay Varshney 180101097
Siddhartha Jain 180101078
Varhade Amey Anant 180101087
Pulkit Changoiwala 180101093

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati

Contents

1 Cohesion and Coupling

1.1 Cohesion

We have divided our code design into five different modules namely User, Toolbox, Gallery, Game and Practice. These modules are chosen to keep in mind high cohesion and low coupling rules for a better software design. To ensure proper cohesion we have chosen functions in each module such that they are sufficient to handle all the needs of that module, as described in the following section:

1. **Module User:** This module caters to all the needs of users.
   In our paint app, we are taking very minimal information from the user, only name and age. Users can see their details and can modify them. Along with this users have the facility to see their saved images and their level in the game mode. For that we have two functions available to users, both this function contributes to the coupling from "Module User" to "Module Gallery" and "Module Game" respectively.

2. **Module Toolbox:** This module contains all the functions to get tools required to draw the image.
   This module provides tools like pencil, eraser, brush along with options to change their size and color. It also offers undo, save and share options for the drawing being drawn thus Toolbox acts as a complete module to handle all the functions required to perform an action on drawing.

3. **Module Gallery:** This module contains all the functions required to handle saved images.
   Various operations can be performed on the saved images like renaming image file, deleting, view, share etc. All the functions are being exercised on image files only thus we introduced this module to have high cohesion(as this module is complete in itself to handle all image related operations) and low coupling(as this module has very minimal dependency on other modules).

4. **Module Game:** This module contains all the functions required to handle game mode of the app.
   Game mode offers functions like selecting a level, submitting the drawing and offering the rewards, thus all these functions are compiled under module Game. Game Mode also requires tools to draw the images thus this module has coupling with Module Toolbox.

5. **Module Practice:** This module contains all the functions required to handle practice mode of the app.
   Practice mode is standalone mode from other modes thus it is obvious to make it a different module. It has no unnecessary dependence on other modules. It has only one function to select a reference image. Practice mode also requires tools to draw the images thus this module has coupling with Module Toolbox.

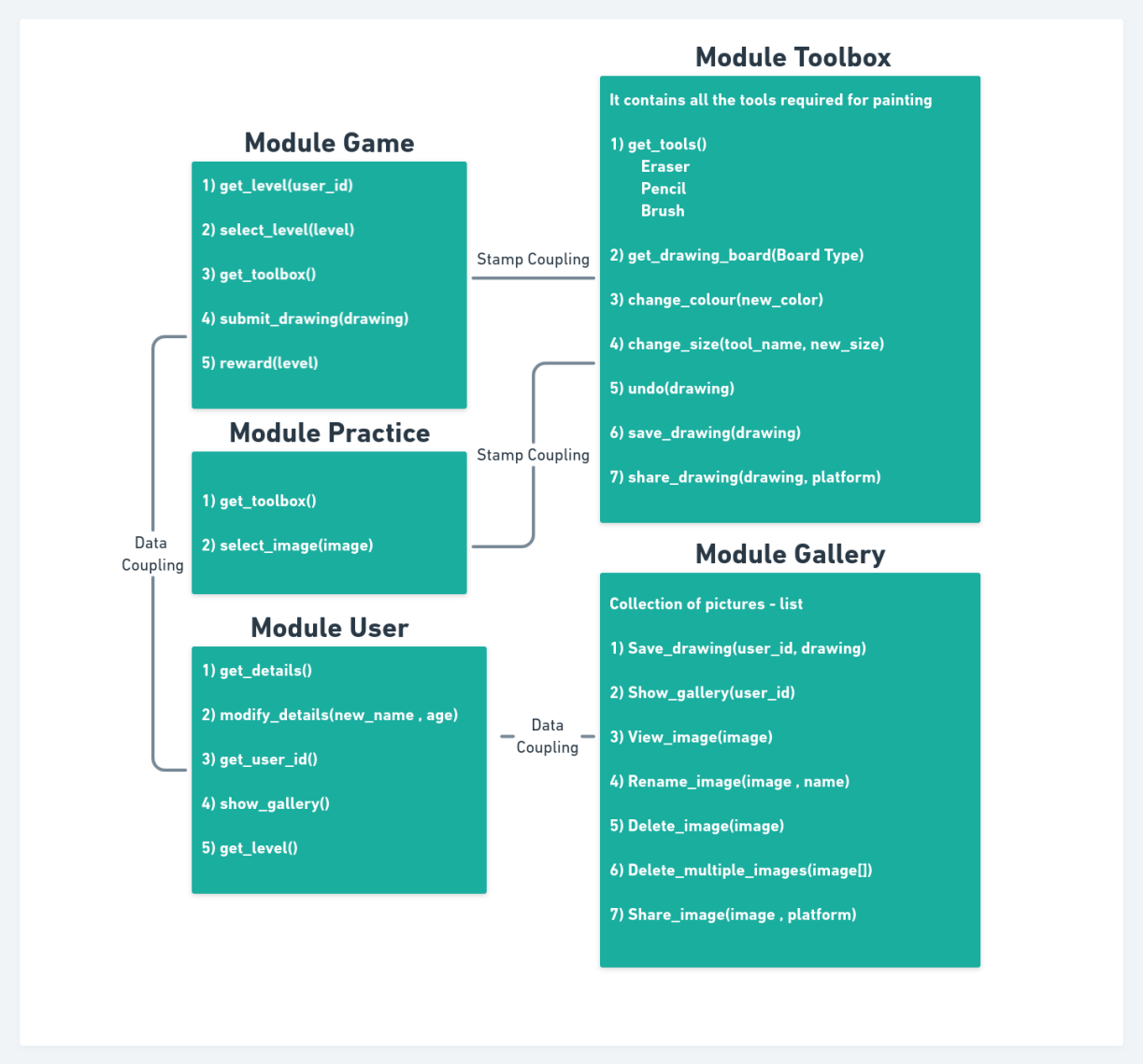

Figure 1.1: Cohesion and Coupling between different modules

1.2 Coupling

We have created modules in such a way to minimize the dependency between modules as much as possible.

1. **Module User:** It is independent of the toolbox module and vice versa. It also doesn't depend upon Practice, Game and Gallery Modules but vice versa is not true.

2. **Module Toolbox:** It is independent of all the other modules present.

3. **Module Gallery:** It depends upon the User Module only via data coupling. While accessing, saving the drawings and some other functions, user id is required which is an attribute of the User module.

4. **Module Game:** It depends upon the User and Toolbox Modules.
   Data Coupling: By using the user id we can get the last unsolved level of that user in the game, issue rewards, jump to previous levels.
   Stamp Coupling: By using the Toolbox Module we can get different tools for drawing in the selected level, saving and sharing the drawing.

5. **Module Practice:** It depends upon the Toolbox Module only via *Stamp Coupling*. While the user is practicing, different tools and their functions are required, saving the image is also needed which are inherited from Toolbox Module.