****

**SSE3306**

**HUMAN COMPUTER INTERFACE**

**SEMESTER 4 2018/2019**

**INDIVIDUAL LAB REPORT**

# TITLE: INVITATION CARD APPLICATION

**NAME: NORAEIN BINTI SABTU**

**MATRIC NO: 192661**

**LECTURER:**  **DR. AZRINA BINTI KAMARUDDIN**

Contents

[TITLE: INVITATION CARD APPLICATION 1](#_Toc8252461)

1. The High Fidelity Prototype…………………………………………………………………………………….2

2. Describing the design with the Nielson Heuristics for UI……………………………………………………..5

3. The rationale of the user interface components selection………………………………………………………6

4. Use case diagram……………………………………………………………………………………………….6

5. The Java Source Code…………………………………………………………………………………………..7

6. Help Description to use Invitation Card application…………………………………………………………..15

7. [Usability Report for The Invitation Card Application. 19](#_Toc8252462)

[1. Executive Summary 19](#_Toc8252463)

[2. Purpose of the Study 19](#_Toc8252464)

[3. Methodology 19](#_Toc8252465)

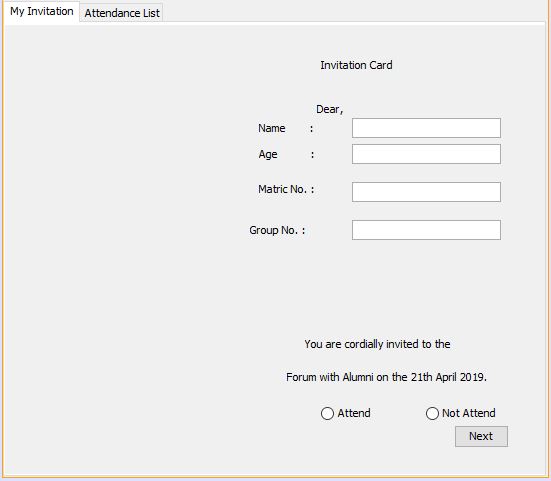
[4. Method 20](#_Toc8252466)

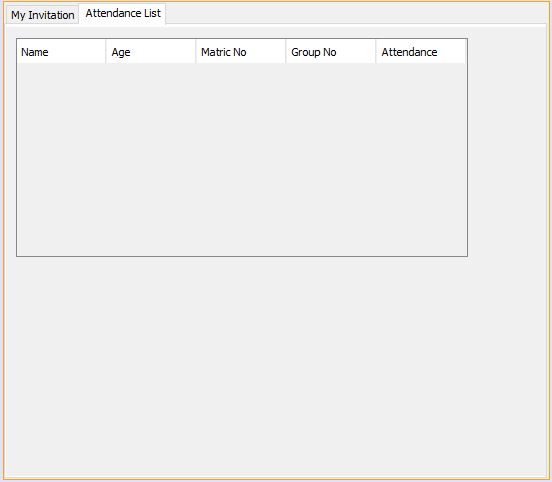
[5. Findings and Recommendations 21](#_Toc8252467)

[6. Discussion 25](#_Toc8252468)

[7. Appendices 25](#_Toc8252470)

1. The High fidelity prototype.





1. Describing the design with the Nielson Heuristics for UI

The user interface for the Invitation Card application that I have developed implement the Jakob Nielsen’s general principles for interaction design. They are called “heuristics” because they are broad rules of thumb and not specific usability guidelines.

The first Heuristics rule that has been applied in my Invitation Card user interface is the visibility of system status. The Invitation Card application always keep users informed about what is going on, through appropriate feedback within reasonable time. For instance, in the Invitation Card application. You can save your details and once you clicked the save button, a popup message will appear in order to notify user that their details has been saved or vice versa.

Second principle is to match between system and the real world. The invitation Card application speak the users’ language with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. In the Invitation Card application, it uses English language so that everyone can understands it better and it uses the tabbed pane in order for the information to appear in a natural and logical order.

Next is user control and freedom where the users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

In the Invitation Card application, the user can choose the emergency exit at the top right of the application and there is also back button option if the user wants to change any of their details that has been input.

The Invitation Card application is very flexible and efficient because it can speed up the interaction for the expert user such as the application can cater both inexperienced and experienced user at once. The application also provide an aesthetic and minimalist design where a lot of dialogue in the application only contain relevant and needed information only. Lastly, the application also help user recognize, diagnose and recover from error. In the Invitation card application, it will notify user if they input a wrong type of data or etc.

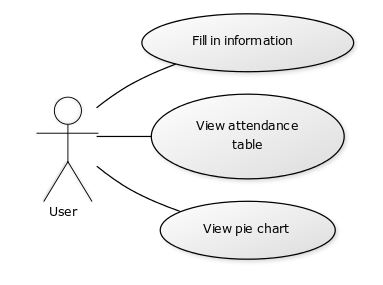
1. The rationale of the user interface components selection.

The invitation card application is using a tabbed pane in order for the user to follow the flow of the application starting by filling in the information about themselves and stating whether they are attending the event or not. Once the user has entered every information that is needed, a button called “Next” is placed at the bottom right of the pane in order for the use to clearly see the button once they have filled in the data that is needed. Radio buttons were used for the attendance section in order for the user to be able to only choose one option whether they are attending or not.

Once the user has filled in their information, it is going to be transferred to the second tab which is called Details confirmation. In this tab, the user shall confirm their details so that the there is no mistake being made regarding the user’s information. There is a back button at the bottom left of the tab in order for the user to go back to the previous tab if there is any changes that they want to make before they proceed to save the information. If there is no changes, user can immediately click the save button at the bottom right of the

Right after the user have saved their details information, there is a button called “list” at the bottom right of the tabbed pane. The button will redirect the user to attendance list table so that the user can see who will be attending the event and vice versa. All the tabbed pane were designed with a light pink background colour in order for the contrast of the content in the application to be visible and clear for the user to view. There are also a few icons being used in order for the application to look way more attractive and does not look too empty.

1. The use case diagram



1. The Java Source code.

**Formula Class**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

// Author: Noraein binti Sabtu

// Matric Number: 192661

// Course code and name: SSE3306 - Human Computer Interface

// Lab Question: An invitation card is sent asking student to input their name to the forum.

// Lecturer: Dr. Azrina binti Kamarudin

// Lab Description: Invitation card application is built that required user to input their details and make confirmation of it. They can also view the list of attendance.

// Objective: Analyze Problem And Construct Program Using GUI Components and applying design principles.

// Last Update: 7/5/2019

package labgui;

/\*\*

\*

\* @author AEIN

\*/

public class formula {

protected String name;

protected int age;

protected String gender;

protected String matricno;

protected int groupno;

protected String attendance;

public formula(){

}

//setter method

public void setName (String name){

this.name= name;

}

public void setAge(int age){

this.age=age;

}

public void setGender(String gender){

this.gender=gender;

}

public void setMatricno(String matricno){

this.matricno=matricno;

}

public void setGroupno(int groupno){

this.groupno=groupno;

}

public void setAttendance(String attendance){

this.attendance=attendance;

}

//getter method

public String getName(){

return name;

}

public int getAge(){

return age;

}

public String getGender(){

return gender;

}

public String getMatricno(){

return matricno;

}

public int getGroupno(){

return groupno;

}

public String getAttendance(){

return attendance;

}

public String toString(){

return "Name : " + getName() +

"\nAge : " + getAge() +

"\nMatric No. : " + getMatricno() +

"\nGroup No. : " + getGroupno() +

"\nAttendance : " + getAttendance();

}

}

**Report Class**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

// Author: Noraein binti Sabtu

// Matric Number: 192661

// Course code and name: SSE3306 - Human Computer Interface

// Lab Question: An invitation card is sent asking student to input their name to the forum.

// Lecturer: Dr. Azrina binti Kamarudin

// Lab Description: Invitation card application is built that required user to input their details and make confirmation of it. They can also view the list of attendance.

// Objective: Analyze Problem And Construct Program Using GUI Components and applying design principles.

// Last Update: 7/5/2019

package labgui;

/\*\*

\*

\* @author AEIN

\*/

public class Report extends formula {

public void ArrayKooooAde(){

// declares an array of integers

int[] anArray;

// allocates memory for 6 integers

anArray = new int[6];

// initialize first element

anArray[0] = 100;

// initialize second element

anArray[1] = 200;

// and so forth

anArray[2] = 300;

anArray[3] = 400;

anArray[4] = 500;

anArray[5] = 600;

System.out.println("Element at index 0: "

+ anArray[0]);

System.out.println("Element at index 1: "

+ anArray[1]);

System.out.println("Element at index 2: "

+ anArray[2]);

System.out.println("Element at index 3: "

+ anArray[3]);

System.out.println("Element at index 4: "

+ anArray[4]);

System.out.println("Element at index 5: "

+ anArray[5]);

}

}

**Main Class**

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

// Author: Noraein binti Sabtu

// Matric Number: 192661

// Course code and name: SSE3306 - Human Computer Interface

// Lab Question: An invitation card is sent asking student to input their name to the forum.

// Lecturer: Dr. Azrina binti Kamarudin

// Lab Description: Invitation card application is built that required user to input their details and make confirmation of it. They can also view the list of attendance.

// Objective: Analyze Problem And Construct Program Using GUI Components and applying design principles.

// Last Update: 7/5/2019

package labgui;

/\*\*

\*

\* @author AEIN

\*/

public class Menu extends javax.swing.JFrame {

/\*\*

\* Creates new form Menu

\*/

public Menu() {

initComponents();

try{

Class.forName("com.mysql.jdbc.Driver");

Connection cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/invitationcard?zeroDateTimeBehavior=convertToNull","root","");

Statement st = cn.createStatement();

JOptionPane.showMessageDialog(null,"Click to enter the Invitation Card Application");

}

catch(Exception e){

JOptionPane.showMessageDialog(null,"Not Connected");

}

}

private void AttendActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

int attend=0;

if(Attend.isSelected()){

attend = attend++;

}

}

private void saveActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

String name = nameTextField.getText();

String age = ageTextField.getText();

String matricno = matricNoTextField.getText();

String groupno = groupNoTextField.getText();

String attendance="";

if(Attend.isSelected()){

attendance =Attend.getText();

}

else if(NotAttend.isSelected()){

attendance =NotAttend.getText();

}

Connection cn = null;

PreparedStatement pt = null;

try{

Class.forName("com.mysql.jdbc.Driver");

cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/invitationcard?zeroDateTimeBehavior=convertToNull","root","");

pt = cn.prepareStatement("insert into myinvitation values (?,?,?,?,?)");

pt.setString(1,name);

pt.setString(2,age);

pt.setString(3,matricno);

pt.setString(4,groupno);

pt.setString(5,attendance);

int i = pt.executeUpdate();

if(i>0){

JOptionPane.showMessageDialog(null,"YOUR RECORD HAS BEEN SAVED! IF YOU WISH TO SEE THE INVITATION GUEST, CLICK THE LIST BUTTON");

}

else{

JOptionPane.showMessageDialog(null,"YOUR RECORD IS NOT SAVED! IF YOU WISH TO SEE THE INVITATION GUEST, CLICK THE LIST BUTTON");

}

}

catch(Exception e){

JOptionPane.showMessageDialog(null,e);

}

}

private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

jTabbedPane1.setSelectedIndex(0);

}

private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

Connection cn = null;

PreparedStatement pt = null;

PreparedStatement pt2 = null;

ResultSet rs = null;

ResultSet rs2 = null;

try{

Class.forName("com.mysql.jdbc.Driver");

cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/invitationcard?zeroDateTimeBehavior=convertToNull","root","");

String sql = "Select count(attendance) from myinvitation where attendance = 'attend'";

String sql2 = "Select count(attendance) from myinvitation where attendance != 'attend'";

pt =cn.prepareStatement(sql);

pt2 = cn.prepareStatement(sql2);

rs= pt.executeQuery();

rs2 = pt2.executeQuery();

if(rs.next() && rs2.next()){

String sum=rs.getString("count(attendance)");

String sum2=rs2.getString("count(attendance)");

DefaultPieDataset pc = new DefaultPieDataset();

pc.setValue("Attend",new Integer(sum));

pc.setValue("Do not Attend", new Integer(sum2));

JFreeChart chart = ChartFactory.createPieChart("Pie Chart", pc, true, true, true);

PiePlot p =(PiePlot)chart.getPlot();

//p.setForegroundAlpha(TOP\_ALIGNMENT);

ChartFrame frame = new ChartFrame("Pie Chart",chart);

frame.setVisible(true);

frame.setSize(450,500);

}

}catch(Exception e){

JOptionPane.showMessageDialog(null, e);

}

String attend=Attend.getText();

String notattend=NotAttend.getText();

}

private void showResult() {

formula p1 = new formula();

p1.setName(nameTextField.getText());

p1.setAge(Integer.parseInt(ageTextField.getText()));

p1.setMatricno(matricNoTextField.getText());

p1.setGroupno(Integer.parseInt(groupNoTextField.getText()));

if(Attend.isSelected()){

p1.setAttendance("Attend");

}

if(NotAttend.isSelected()){

p1.setAttendance("Do not Attend");

}

//Attendance table

if(p1.getAttendance()== "Attend"){

FinalReportTextArea.setText(p1.toString());

}

else{

FinalReportTextArea.setText(p1.toString());

}

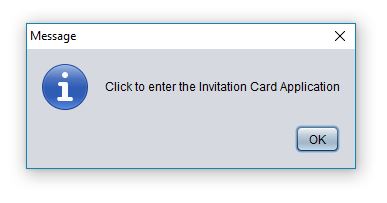
//final report

}

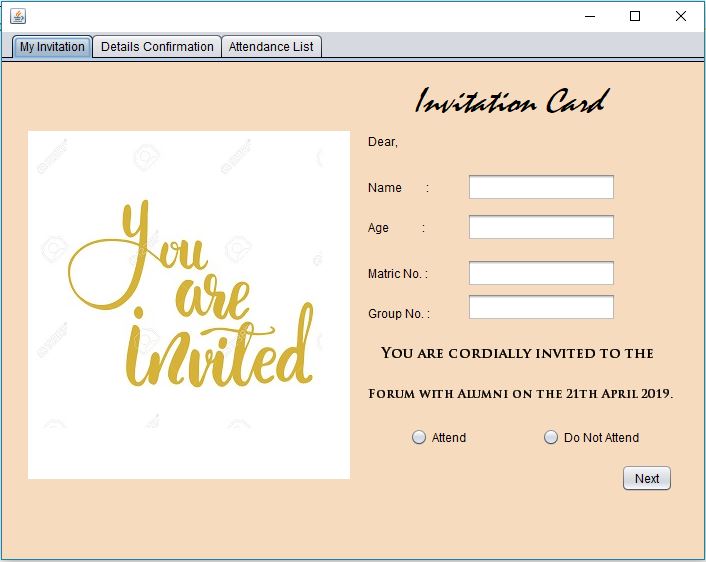
}

1. Help Description to use Invitation Card application

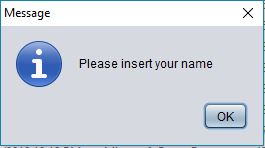
The Invitation Card application starts with a popup message that notify the user that they are about to enter the application.



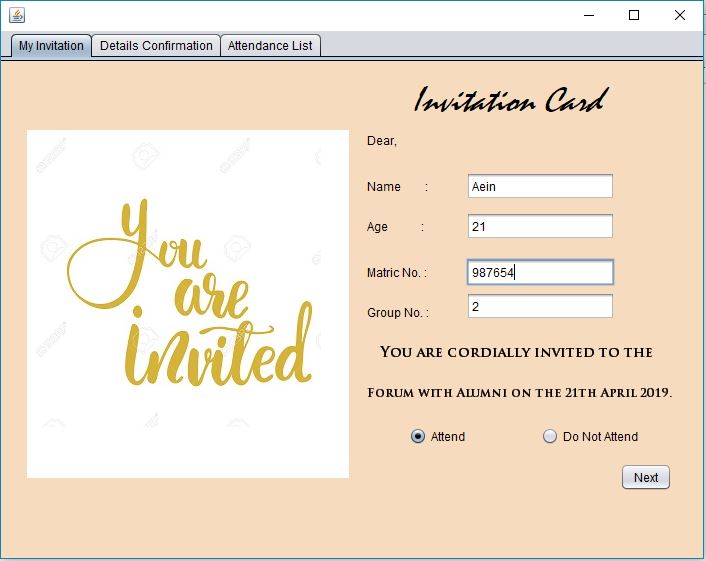
Once the user pressed the “OK” button, user will be directed to the Invitation Card application first interface.



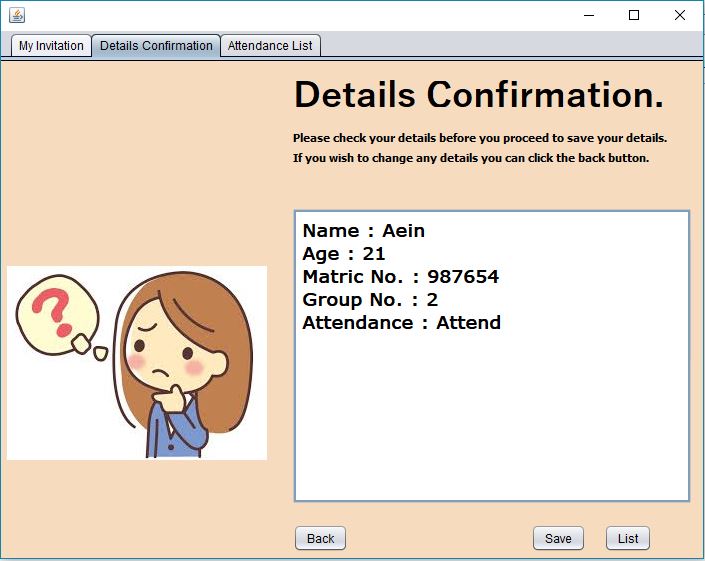
In the first interface of the application, user is supposed to fill in the data that are needed such as name, matric no and etc. If the user didn’t enter any data, a message will be popped up indicating that the field cannot be empty.



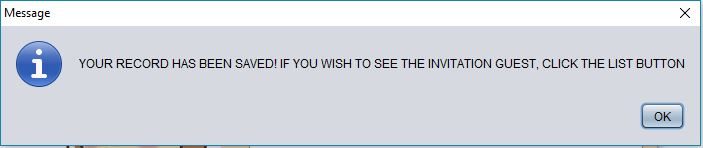
Once the user has filled in the details, the user can proceed by clicking the “Next” button at the bottom right of the tabbed pane.



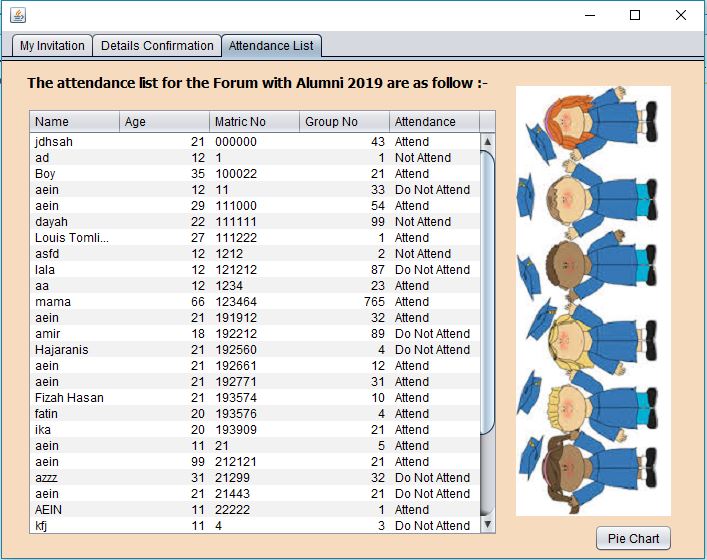
The user will be directed to the second tab which is the details confirmation tab where the user can check their details whether it is correct or not. If there is any changes that they want to make, the user can just click the “Back” button at the bottom left of the tab so it will redirect the user to the first tab so that the user can modify their information. If all the details are correct, the user can proceed and click the “Save” button in order to save their details.



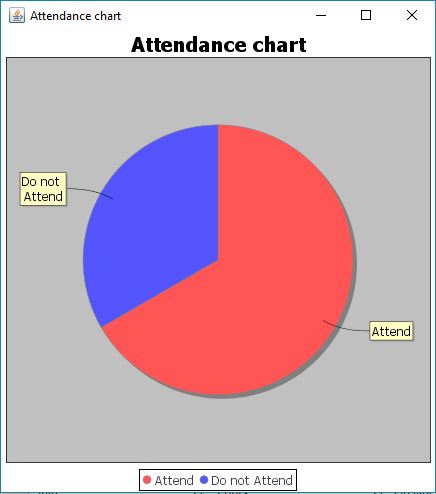
A popup message will appear indicating whether the details is saved or vice versa. In the popup message, the user were informed if they want to view the attendance list, they can simply click the “List” button so that the user will be redirected to the third tab which is the Attendance list tab.



The attendance list interface are as follows. The user can view who will and will not attend the event.



User also have another option of viewing the attendance list where the user can view the pie chart or the attendance list that shows the chart of who is going to attend and not going to attend. The pie chart can be retrive when the user click the “Pie chart” button.



# Usability Report for The Invitation Card Application.

# 1. Executive Summary

The usability testing regarding the Invitation Card application that has been done by using a questionnaire. The purpose of conducting the usability testing is to see what the users feel about the Invitation card application that includes question that test the user’s satisfaction, learnability, ease of use and etc. The test were conducted for a span of a week and the number of participants were 5 peoples. The test includes 9 questions where all the questions test the user’s impression while using the application. All in all, I can conclude that, from the findings through the questionnaire that the respondents are very satisfied with the Invitation Card application because it satisfied the users’ requirements.

# 2. Purpose of the Study

The purpose or objective of this usability testing is to test on how the user feels while using the Invitation card application. Every user might have a different or even the same opinion towards the application. In order to know the point of view of the user, this usability testing was conducted so that we can know what they really feel while browsing through the Invitation card application.

**The goals of the usability testing is to check whether the participants think that the Invitation Card application is easy to use or they find it hard to navigate around it. The application also want to ensure whether the participants feel satisfied or not while using it and whether it is easy or hard for the participants to learn on how to use the application if it is their first time using it.**

# 3. Methodology

The participants for the usability testing were chosen randomly because the application that I am working on is an application that can be use by anyone. The total number of participants that were involved in the test were 5 people. The age of the participants were asked at the beginning of the questionnaire in order for me to know what range of age does the participants from.

# 4. Method

Questionnaire were used in order to record the participants’ answers. The questionnaire were taken from <https://it.toolbox.com/blogs/craigborysowich/sample-application-usability-test-questionnaire-072407> .

At the beginning of the questionnaire, the participants were asked to use the Invitation Card application so that the user can experience how the application works. After the participants has finished using the application, they can start to answer the questionnaire. The questionnaire consists of 9 questions that includes questions regarding learnability, satisfaction and ease of use of the application.

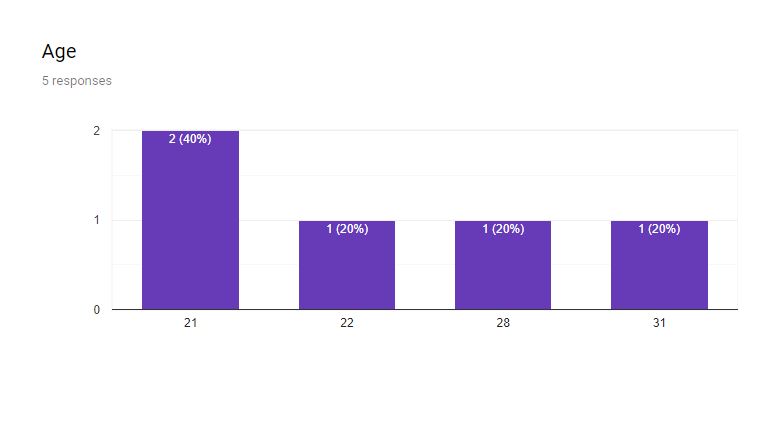
The first information that was recorded was the age of the participants. The age of the participants was recorded in order for us to know what range of age did the respondents were categorized in. The second question was asking the participants on how they felt about the Invitation Card application, is it easy to use or it is very difficult for them. The third question was about whether the menu and functions on the application are organized and easy to use or not.

The fourth question is to test the learnability aspect of the Invitation Card application. The question that were asked was whether the participants immediately understands how to use the application or not. The next question correlates with the ease of use of the application where the participants were asked whether the buttons on the application are well organized and easy to find. The participants were asked on how they feel while navigating around the app, whether it is very difficult or not.

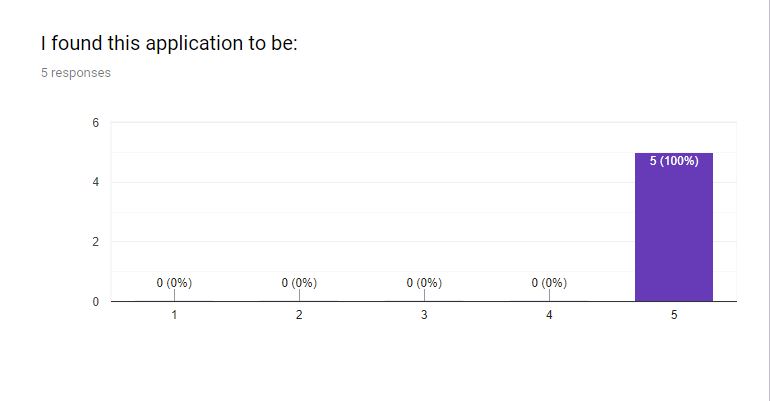
The participants were also asked about the complexity of the Invitation Card application and they can choose whether they feel it is very complex or vice versa. The last question is for the participants to express what their overall impression towards the Invitation Card application that has been created and they can write their comment as they wish.

# 5. Findings and Recommendations

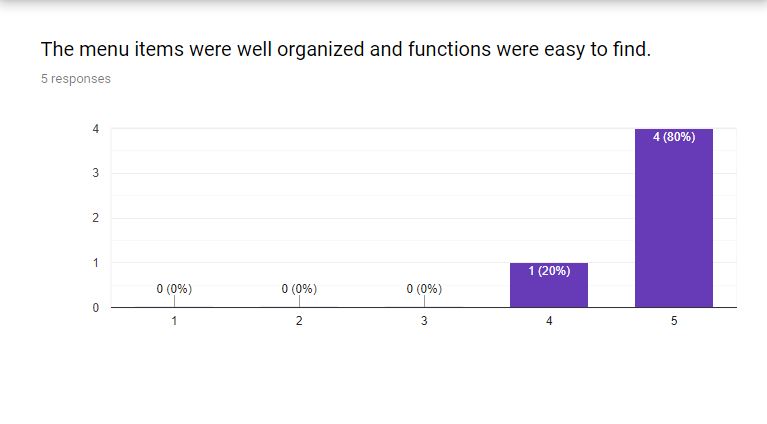
Findings from the questionnaire.



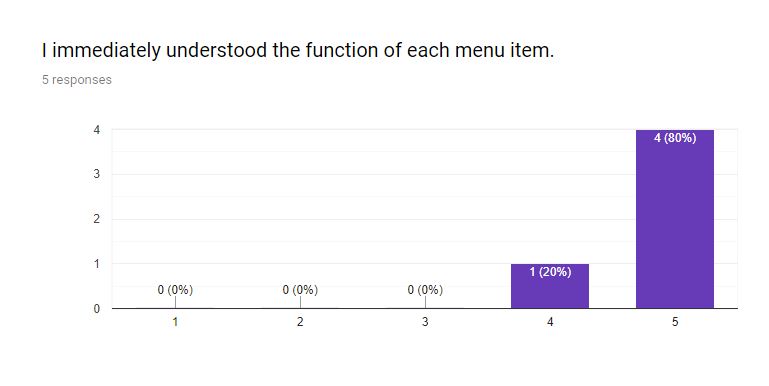
The age of the participants are in between 21-31.



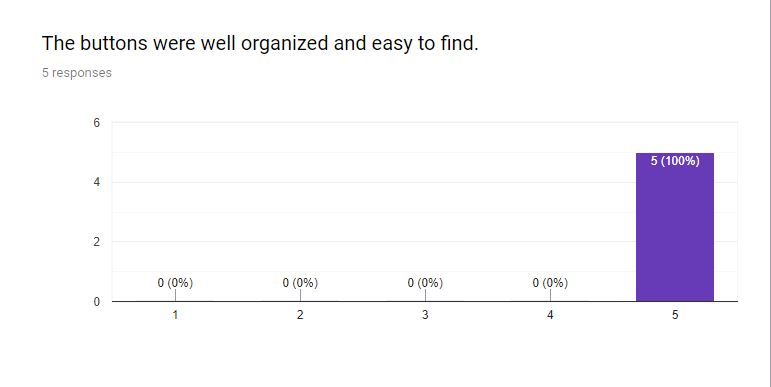
Five out of five of the respondents find that the wireframe was extremely easy to use.



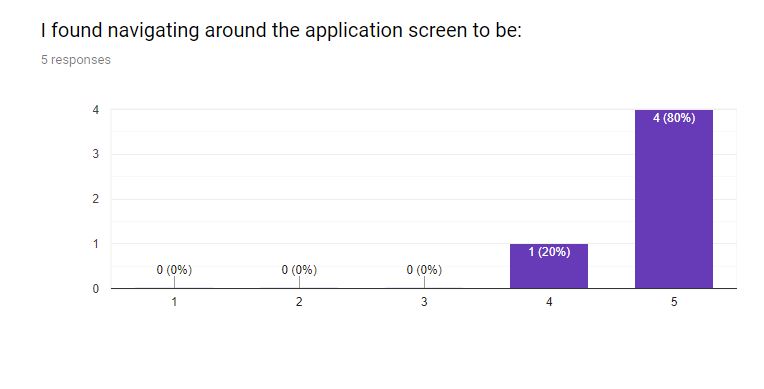
Four out of five of the participants strongly agrees that the menu items were very organized and the functions were very easy to find while one of the participant agrees that the menu items were very organized and the functions were very easy to find.



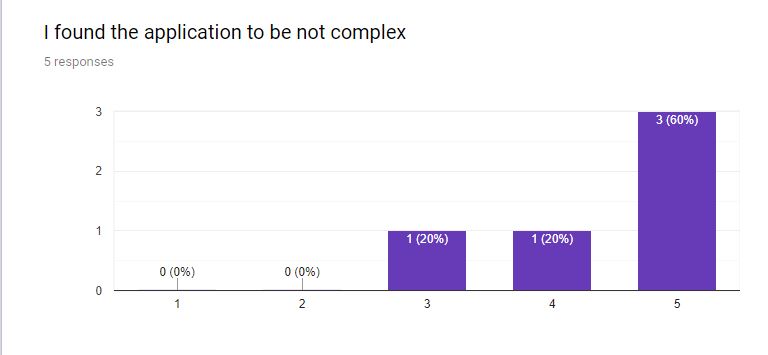
Four out of five of the respondents strongly agrees that they immediately understood the function of each menu and only one of the respondent finds that they understood the function of each menu item.



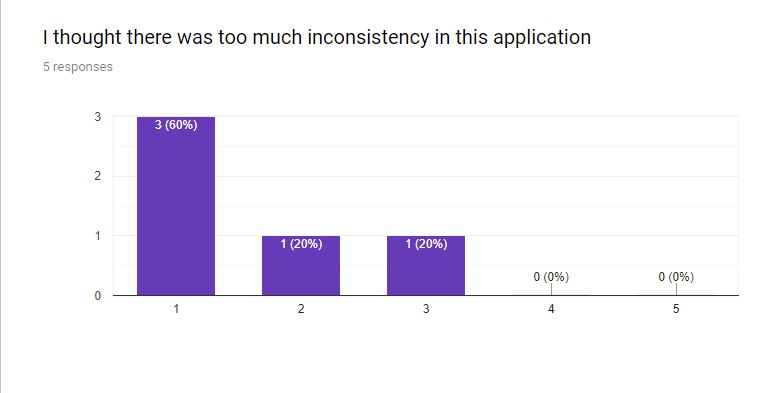
All of the respondents strongly agrees that the buttons on the wireframe were well organized and easy to find.



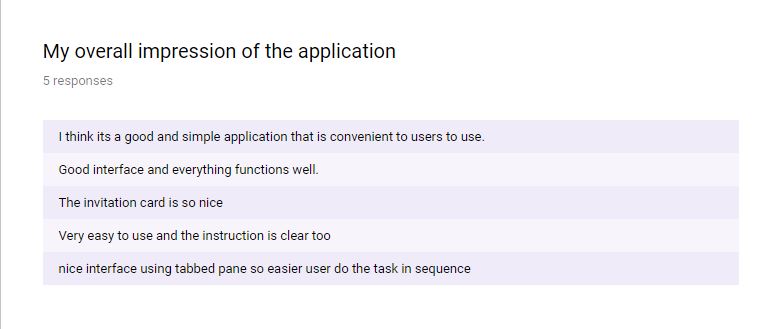
Only one out of five of the respondents agrees it was easy to navigate around the wireframe screen while four of the respondents strongly agrees that it was very easy to navigate around the wireframe screen.



Three out of five of the respondents strongly agrees that the application was not complex, one of the respondent feel that the application complexity is average and the last respondent agree that the application is not complex.



Three out of five of the respondents strongly disagree that there was too much inconsistency in the application, one of the respondent disagree that there was too much inconsistency in it and only one respondent feel moderately about the inconsistency of the Invitation Card application.



The last question from the questionnaire shows the respondents overall impression towards the Invitation Card application.

# 6. Discussion

# All in all, I can say that the Invitation Card application has been designed according to how the respondents want it to be.

**Most of the participants finds that t**he application is very organised and easy to navigate around the application. The information inside the application is very clear that makes the user has no problem while navigating around the application. It also shows that it is very easy for a first time user to use the application because all of them seems to not have any problem while using the application. The usage of tabbed pane makes the user to complete the task easily in sequential manner. Overall the respondents were really satisfied with the Invitation Card application.

# 7. Appendices