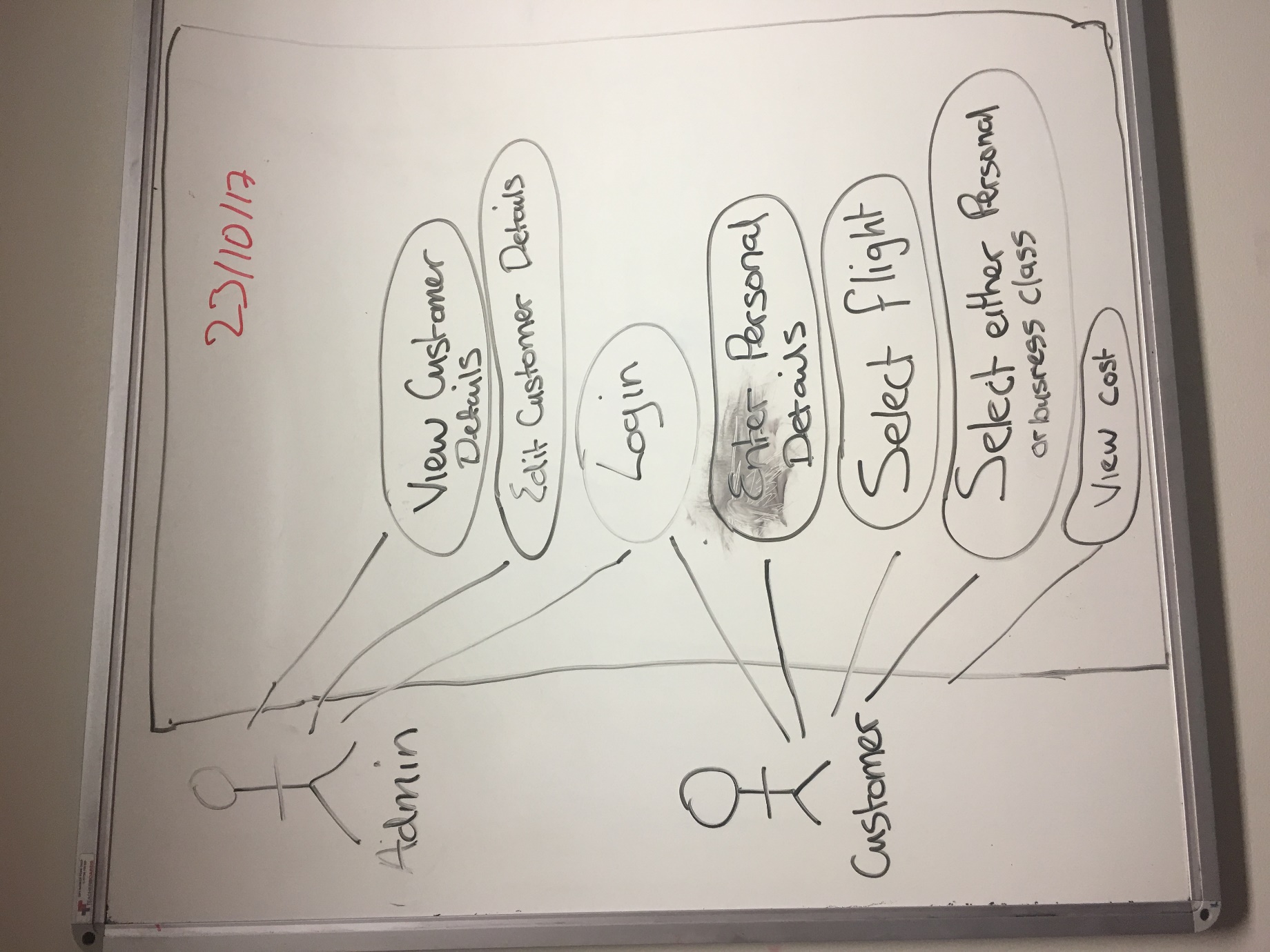
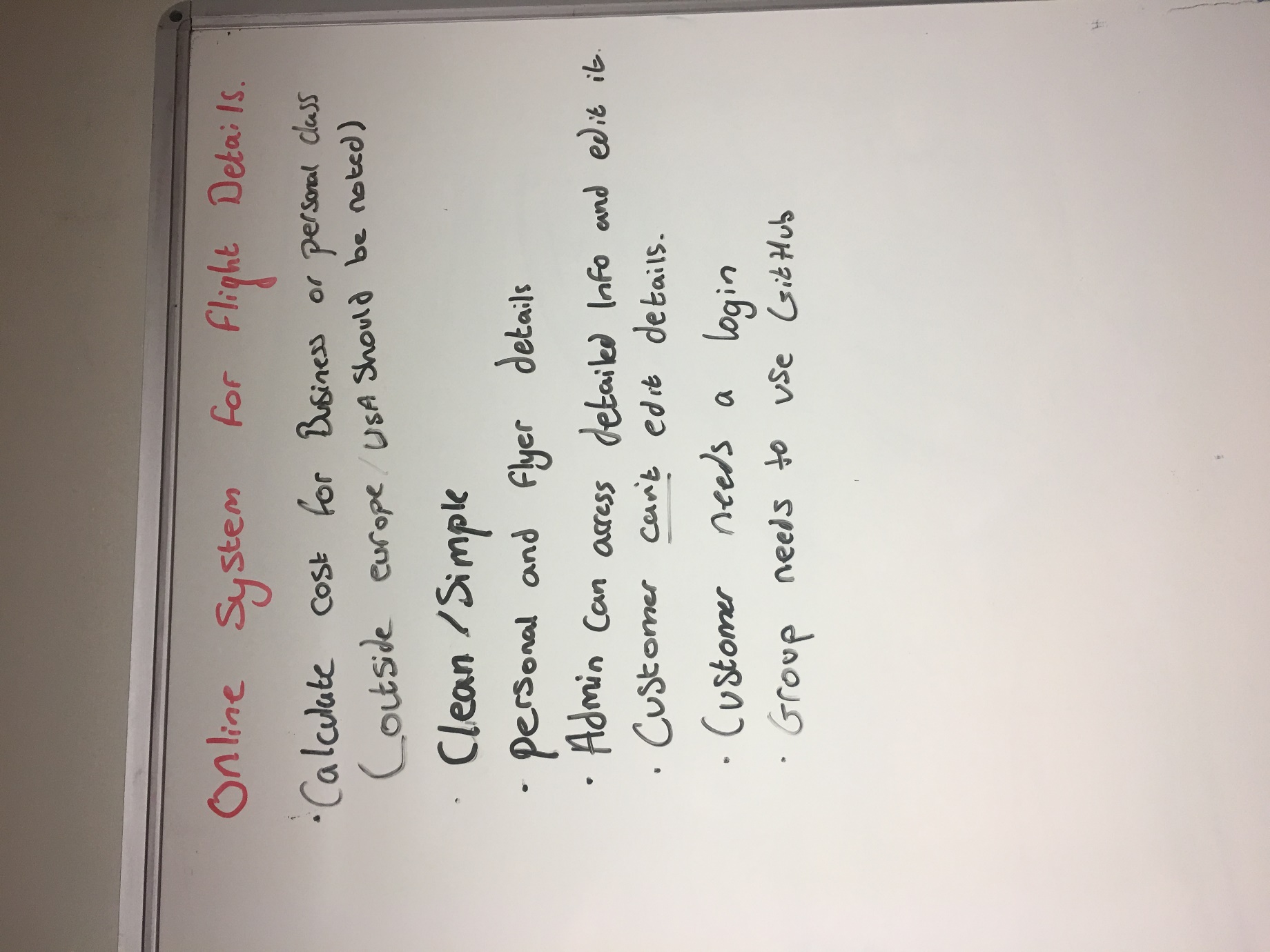
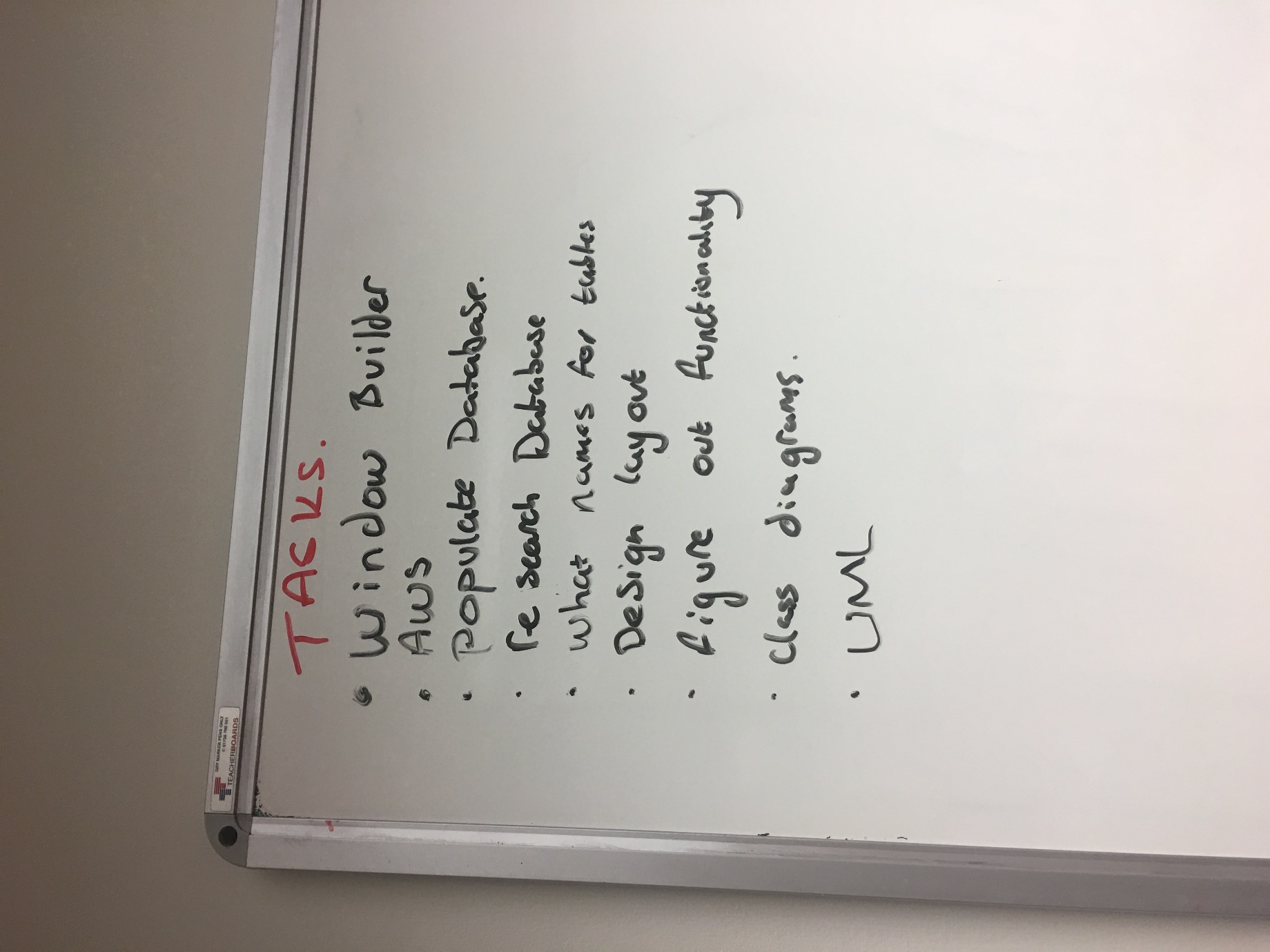
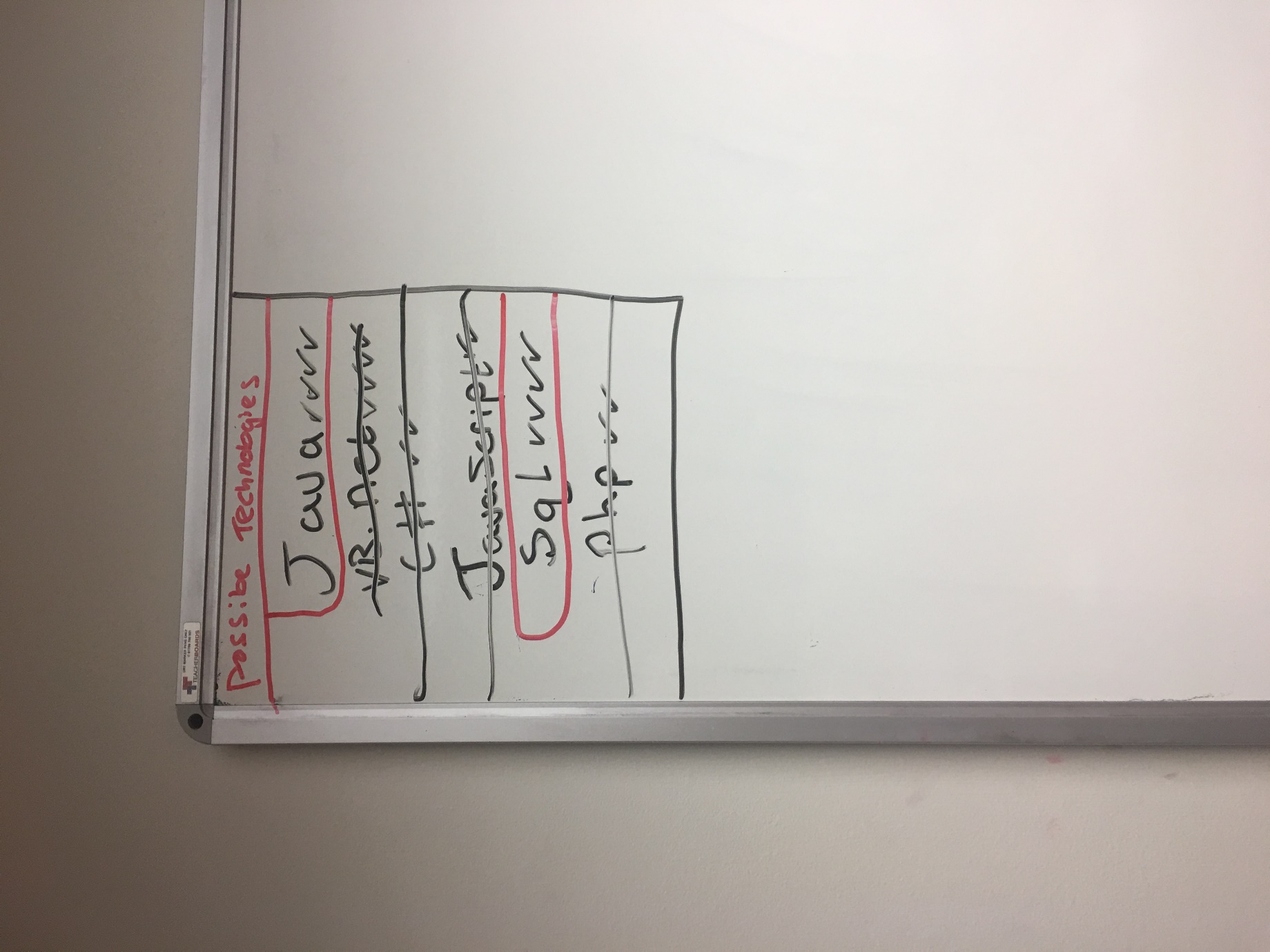
# Introduction

# Images from first meeting



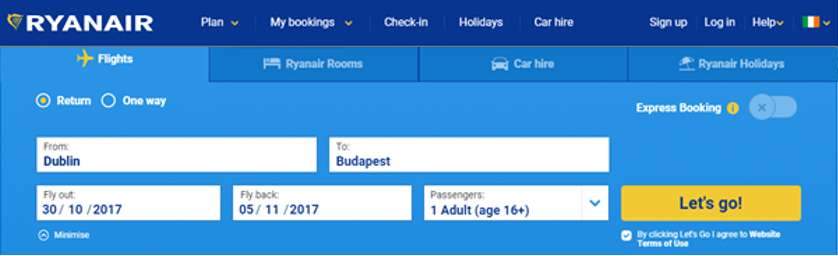
# Diagrams

# Business Requirements

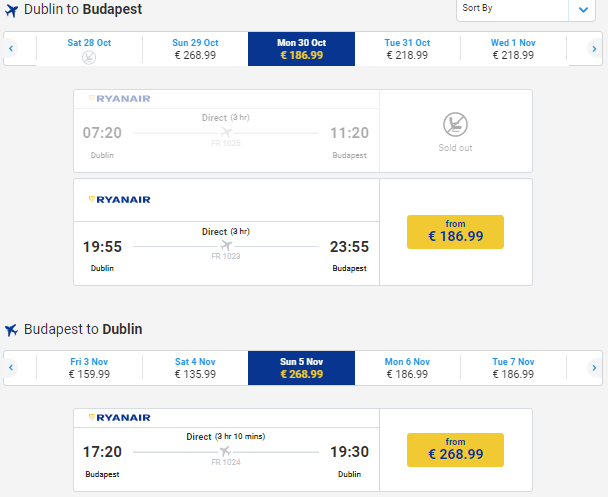
\*\*Define business requirements here\*\*\*

# Review of Existing Systems

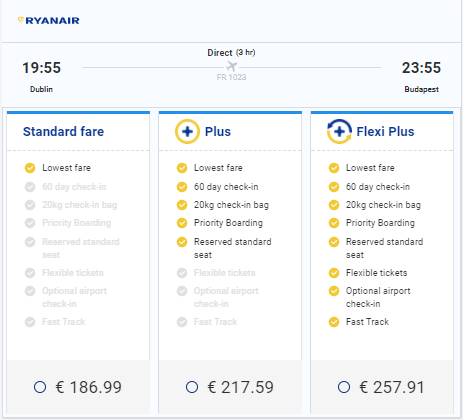
This section samples an existing system which serves a similar purpose to the system we have been assigned for our team project. It has been examined to so that we can create our own GUI that works in a similarly efficient way.



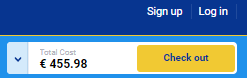
### **Figure 1.1:** Ryanair’s Initial Menu



### **Figure 1.2:** Price for flight(s) displayed.



### **Figure 1.3:** Additional options displayed.



### **Figure 1.4:** Cost displayed with no user login required.

Our system will be developed in Java using eclipse. The GUI above can be similarly represented on a Java GUI platform as the Java components that exist are quite similar to this web based system’s components (eg JButton, JTextfield, JCheckBox, JDialog, JRadioButton etc.).

# Jacoco

# Junit/TestNG

# Jira

# JConsole/jRAT

# Javadoc

# Code Created

## Consideration for security of code

## Consideration for performance of code

# Team Collaboration

## Agile Development

## GIT

# Database

## Amazon RDS Cloud Database Connect On SQLWorkBench

1. Add connection
2. Hostname: devops.clql55s9fxrz.eu-west-1.rds.amazonaws.com
3. Username: DevOps
4. Password: groupthree

Test connection and should work, my connection name is DevOps but don’t think it matters what you call it, I added some sample tables.

## Some Sample Java Code to Connect to Database

/\*\*

\* Database handler

\*

\* @author Mark Glenn

\*/

import javax.swing.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

@SuppressWarnings("serial")

public class DatabaseHandler extends javax.swing.JFrame {

// credentials for database including AWS RDS database endpoint and JDBC

// driver

final String JDBC\_DRIVER = "com.mysql.jdbc.Driver";

final String DB\_URL = " devops.clql55s9fxrz.eu-west-1.rds.amazonaws.com";

final String USER\_NAME = "cloud1";

final String PASSWORD = "211230mg";

Connection conn = null;

Statement stmt = null;

ResultSet rs = null;

// connect to database

public void connectToDatabase() {

try {

// STEP 1 - Load the JDBC driver

java.lang.Class.forName(JDBC\_DRIVER);

System.out.println("STEP 1 COMPLETE - Driver Registered...");

// STEP 2 - Open a connection

conn = DriverManager.getConnection(DB\_URL, USER\_NAME, PASSWORD);

System.out.println("STEP 2 COMPLETE - Connection obtained...");

// STEP 3 - Create Statement object

stmt = conn.createStatement();

System.out.println("STEP 3 COMPLETE - Statement object created...");

} catch (ClassNotFoundException e) {

System.out.print("Connection Error");

//JOptionPane.showMessageDialog(null,"Could not load driver.\n" + e.getMessage());

} catch (SQLException e) {

System.out.print("Connection Error");

//JOptionPane.showMessageDialog(null,"Problem with SQL.\n" + e.getMessage());

}

}

// connect and do query

public void doQuery(String query) {

try {

java.lang.Class.forName(JDBC\_DRIVER);

System.out.println("STEP 1 COMPLETE - Driver Registered...");

// STEP 1 - Open a connection

conn = DriverManager.getConnection(DB\_URL, USER\_NAME, PASSWORD);

System.out.println("STEP 2 COMPLETE - Connection obtained...");

// STEP 2 - Create Statement object

stmt = conn.createStatement();

System.out.println("STEP 3 COMPLETE - Statement object created...");

System.out.println("STEP 4(a) COMPLETE - Query executed and database found...");

System.out.println("STEP 4(b) COMPLETE - Query executed.");

} catch (Exception ex) {

JOptionPane.showMessageDialog(null, ex.getMessage());

}

}

}// end of class

## Database persistence technology

# Conclusions