ISAD251 Report

APEX811: Cody Robinson, Reece Davies,

Jaime Kershaw Brown, Erin Clifton

Plymouth University

Nov 2018

Contents

Introduction

1. Scenario

There is a need for a university system that allows stage-two computing students to apply for a placement year. This application will be tracking the student's activity throughout the application process, from the start of stage-two until they have secured a placement. It will not focus solely on students, but also be capable of managing companies and the jobs available to students. Students will be capable of applying through a mobile app, whilst the university will be able to manage students and companies through the admin desktop app.

2. Background information

Relating specifically to the requirements of the system, the admin will be required to input the students' information in order for them to use the system. This includes: student ID (which is auto generated), the code of the programme they are undertaking, full name, email address, mobile and landline telephone numbers, date of birth, home and term time address. Furthermore, specific values must not be null and in the format that best suites the content. In order for a student to apply for a job, they are required to submit a CV and have it approved by the admin. Once this has been accomplished, they will be deemed 'active' and can start applying for jobs.

A company is capable of having one or more sites; the jobs provided by a company will be related to a specific site the company owns, and thus will be the location for the job. This includes the address and postcode. In order for the student to apply for a job, they must be informed about the specifics of the job, and therefore the job must include: a job title, a short description outlining the job, contact information for enquiries, the salary of the job, the start date for the placement, a closing date for applications and an application method. Specific jobs will also require information relating to the student's application history, which will outline the dates and times of the student's status throughout their previous applications. The admin will be responsible for adding companies, sites and jobs.

3. Assumptions made

- Each site will have its own independent email address and telephone number for students to contact enquiring about the job available.
- Each site can have multiple jobs.
- Each job can have multiple vacancies available.
- Each company should have at least one site.
- There is no limit for how many sites a company can have.
- Admin will be responsible for creating the students' accounts.
- A company's name will not change.
- Once a student has accepted a placement they shall make no further applications.
- Companies will not close down sites during the application process.
- The company will make their choice of applicant for a job after the closing date.
- A student will not make multiple applications to the same job.
- A student will not be entered into the system twice.
- A student can only be at each stage in the application process once

4. Functional requirements

Admin Desktop Application

Students
- Add a student to the database
- View a student’s information
- View lists of both active & inactive students
- View a list of active students who have not yet accepted an offer
- View a list of students who have accepted a placement offer
- View a report of a student’s application(s) status
- View a pie chart showing the proportion of placed to unplaced students
- View a stacked bar chart showing the number of active students placed & unplaced by programme
- View a calendar of job vacancy closing dates
- Amend student information

Companies
- Add a company to the database
- View a company’s information
- Update a company’s information

Job Vacancies
- Add a job vacancy
- View an individual job
- View a list of jobs (current & past job vacancies)
- View a list of jobs that close within the next 7 days
- Update a given job vacancy

Student Mobile Application

Jobs

- View a list of current jobs (current & past job vacancies)

Applications

- Record an application made
- Add a new status of application
- View a report on the status of all applications and each application made

Students

- View a student’s information including
- Amend student information

5. Project plan

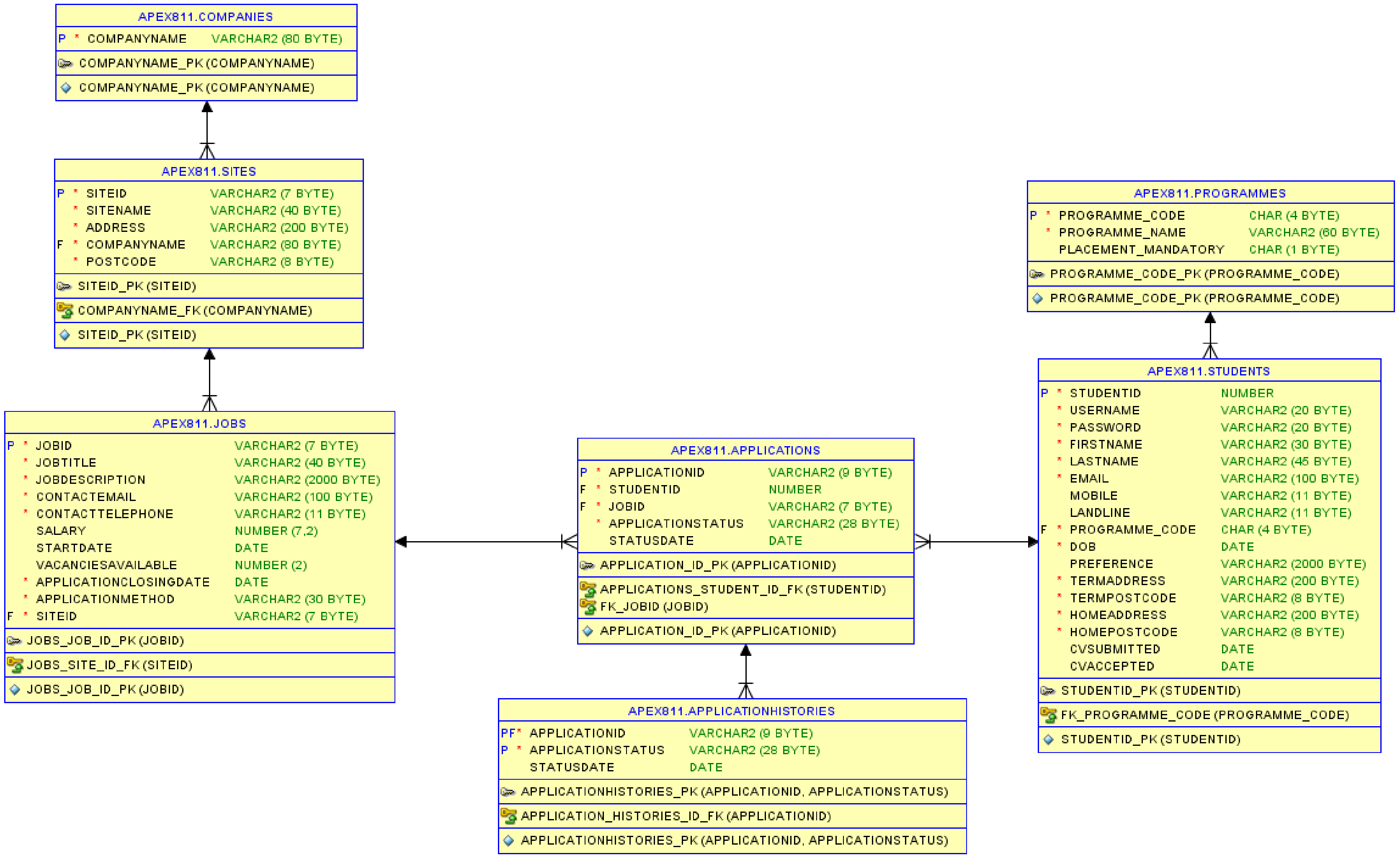
Activity	Expected Start Date	Estimate Hours (Per Person)	Actual Start date	Actual Hours spent (Per Person)
Read through of Assignment brief, and discussion.	22/10/2018	2	18/10/2018	2
Create Project Plan for assignment.	24/10/2018	5	31/10/2018	4
Listed all the functional requirements.	26/10/2018	2	23/10/2018	5
List all attributes.	29/10/2018	1	19/10/2018	1
Normalise attributes into third normal form, alongside with declaring their table name and their data types.	31/10/2018	6	19/10/2018	8
Create relevant ERD for normalised data.	06/11/2018	4	22/10/2018	3
Design storyboard for front-end view of system.	09/11/2018	3	31/10/2018	6
Create SQL scripts to generate all tables in Oracle, alongside with constraints to relevant primary, composite and foreign keys.	14/11/2018	5	29/10/2018	7
Popularised the database with relevant information.	23/11/2018	4	26/11/2018	4
Create graphical user interface for front-view.	27/11/2018	9	27/11/2018	17
Create Report to review Assignment.	07/12/2018	9	03/12/2018	12

Development

1. Method of Approach

Once we had our four members for the group, we arranged a meeting in which we read through the assignment brief and discussed which tasks should be performed first with the best method to do so. Following this we started naming all the attributes that the system might contain, including the ones stated in the assignment brief. This may have not been the best approach as we did not initially create a project plan prior to starting the coursework. As a result, we were required to spend more time planning on what actions were left to do. This is because we were unsure of which tasks were complete, semi-complete or incomplete. Once the project plan was created, it acted as a foundation of what task must be worked upon, when it must be started, and the time spent on that specific task. Of course, certain tasks had complications and thus the actual amount of hours spent on it exceeded the estimate amount of hours. This was particularly problematic when developing the graphical user interface with APEX as we were unfamiliar with the software.

2. Entity Relationship Diagram



APEX811.COMPANIES

P	*	COMPANYNAME	VARCHAR2 (80 BYTE)

COMPANYNAME_PK (COMPANYNAME)
COMPANYNAME_PK (COMPANYNAME)

APEX811.SITES

P	*	SITEID	VARCHAR2 (7 BYTE)
	*	SITENAME	VARCHAR2 (40 BYTE)
	*	ADDRESS	VARCHAR2 (200 BYTE)
F	*	COMPANYNAME	VARCHAR2 (80 BYTE)
	*	POSTCODE	VARCHAR2 (8 BYTE)

SITEID_PK (SITEID)
COMPANYNAME_FK (COMPANYNAME)
SITEID_PK (SITEID)

APEX811.JOBS

P	*	JOBID	VARCHAR2 (7 BYTE)
	*	JOBTITLE	VARCHAR2 (40 BYTE)
	*	JOBDESCRIPTION	VARCHAR2 (2000 BYTE)
	*	CONTACTEMAIL	VARCHAR2 (100 BYTE)
	*	CONTACTTELEPHONE	VARCHAR2 (11 BYTE)
		SALARY	NUMBER (7,2)
		STARTDATE	DATE
		VACANCIESAVAILABLE	NUMBER (2)
	*	APPLICATIONCLOSINGDATE	DATE
	*	APPLICATIONMETHOD	VARCHAR2 (30 BYTE)
F	*	SITEID	VARCHAR2 (7 BYTE)

JOBS_JOB_ID_PK (JOBID)
JOBS_SITE_ID_FK (SITEID)
JOBS_JOB_ID_PK (JOBID)

APEX811.APPLICATIONS

P	*	APPLICATIONID	VARCHAR2 (9 BYTE)
F	*	STUDENTID	NUMBER
F	*	JOBID	VARCHAR2 (7 BYTE)
	*	APPLICATIONSTATUS	VARCHAR2 (28 BYTE)
		STATUSDATE	DATE

APPLICATION_ID_PK (APPLICATIONID)
APPLICATIONS_STUDENT_ID_FK (STUDENTID)
FK_JOBID (JOBID)
APPLICATION_ID_PK (APPLICATIONID)

APEX811.APPLICATIONHISTORIES

PF	*	APPLICATIONID	VARCHAR2 (9 BYTE)
P	*	APPLICATIONSTATUS	VARCHAR2 (28 BYTE)
		STATUSDATE	DATE

APPLICATIONHISTORIES_PK (APPLICATIONID, APPLICATIONSTATUS)
APPLICATION_HISTORIES_ID_FK (APPLICATIONID)
APPLICATIONHISTORIES_PK (APPLICATIONID, APPLICATIONSTATUS)

APEX811.PROGRAMMES

P	*	PROGRAMME_CODE	CHAR (4 BYTE)
	*	PROGRAMME_NAME	VARCHAR2 (60 BYTE)
		PLACEMENT_MANDATORY	CHAR (1 BYTE)

PROGRAMME_CODE_PK (PROGRAMME_CODE)
PROGRAMME_CODE_PK (PROGRAMME_CODE)

APEX811.STUDENTS

P	*	STUDENTID	NUMBER
	*	USERNAME	VARCHAR2 (20 BYTE)
	*	PASSWORD	VARCHAR2 (20 BYTE)
	*	FIRSTNAME	VARCHAR2 (30 BYTE)
	*	LASTNAME	VARCHAR2 (45 BYTE)
	*	EMAIL	VARCHAR2 (100 BYTE)
		MOBILE	VARCHAR2 (11 BYTE)
		LANDLINE	VARCHAR2 (11 BYTE)
F	*	PROGRAMME_CODE	CHAR (4 BYTE)
	*	DOB	DATE
		PREFERENCE	VARCHAR2 (2000 BYTE)
	*	TERMADDRESS	VARCHAR2 (200 BYTE)
	*	TERMPOSTCODE	VARCHAR2 (8 BYTE)
	*	HOMEADDRESS	VARCHAR2 (200 BYTE)
	*	HOMEPOSTCODE	VARCHAR2 (8 BYTE)
		CVSUBMITTED	DATE
		CVACCEPTED	DATE

STUDENTID_PK (STUDENTID)
FK_PROGRAMME_CODE (PROGRAMME_CODE)
STUDENTID_PK (STUDENTID)

3. SQL Scripts

```
CREATE TABLE "APEX811"."COMPANIES"
   ( "COMPANYNAME" VARCHAR2(80 BYTE) CONSTRAINT "COMPANYNAME_NN" NOT NULL ENABLE,
     CONSTRAINT "COMPANYNAME_PK" PRIMARY KEY ("COMPANYNAME")
   );

CREATE TABLE "APEX811"."SITES"
   ( "SITEID" VARCHAR2(7 BYTE) DEFAULT ON NULL
'S'||LTRIM(TO_CHAR("APEX811"."SEQ_SITE_ID"."NEXTVAL",'09999')) NOT NULL ENABLE,
     "SITENAME" VARCHAR2(40 BYTE) CONSTRAINT "SITENAME_NN" NOT NULL ENABLE,
     "ADDRESS" VARCHAR2(200 BYTE) CONSTRAINT "ADDRESS_NN" NOT NULL ENABLE,
     "COMPANYNAME" VARCHAR2(80 BYTE) CONSTRAINT "SITE_COMPANYNAME_NN" NOT NULL ENABLE,
     "POSTCODE" VARCHAR2(8 BYTE) CONSTRAINT "SITE_POSTCODE_NN" NOT NULL ENABLE,
      CONSTRAINT "SITEID_PK" PRIMARY KEY ("SITEID"),
      CONSTRAINT "SITE_POSTCODE_CHK" CHECK (postcode LIKE '%___ ___') ENABLE,
      CONSTRAINT "COMPANYNAME_FK" FOREIGN KEY ("COMPANYNAME")
       REFERENCES "APEX811"."COMPANIES" ("COMPANYNAME") ENABLE
   ) ;

CREATE SEQUENCE  "APEX811"."SEQ_SITE_ID"  MINVALUE 1 MAXVALUE 99999 INCREMENT BY 1 START WITH 10016
NOCACHE  NOORDER  CYCLE  NOKEEP  NOSCALE  GLOBAL ;
```

```
CREATE TABLE "APEX811"."JOBS"
   (  "JOBID" VARCHAR2(7 BYTE) DEFAULT ON NULL
'J'||LTRIM(TO_CHAR("APEX811"."SEQ_JOB_ID"."NEXTVAL",'09999')) NOT NULL ENABLE,
      "JOBTITLE" VARCHAR2(40 BYTE) CONSTRAINT "JOBS_JOB_TITLE_NN" NOT NULL ENABLE,
      "JOBDESCRIPTION" VARCHAR2(2000 BYTE) CONSTRAINT "JOBS_JOB_DESCRIPTION_NN" NOT NULL ENABLE,
      "CONTACTEMAIL" VARCHAR2(100 BYTE) CONSTRAINT "JOB_EMAIL_NN" NOT NULL ENABLE,
      "CONTACTTELEPHONE" VARCHAR2(11 BYTE) CONSTRAINT "JOB_TELEPHONE_NN" NOT NULL ENABLE,
      "SALARY" NUMBER(7,2),
      "STARTDATE" DATE,
      "VACANCIESAVAILABLE" NUMBER(2,0),
      "APPLICATIONCLOSINGDATE" DATE CONSTRAINT "JOBS_APPLICATION_CLOSING_DATE_NN" NOT NULL ENABLE,
      "APPLICATIONMETHOD" VARCHAR2(30 BYTE) CONSTRAINT "JOBS_APPLICATION_METHOD_NN" NOT NULL ENABLE,
      "SITEID" VARCHAR2(7 BYTE) CONSTRAINT "SITEID_NN" NOT NULL ENABLE,
       CONSTRAINT "JOBS_JOB_ID_PK" PRIMARY KEY ("JOBID"),
       CONSTRAINT "JOB_EMAIL_CHK" CHECK (contactemail LIKE  '%_@___%.__%') ENABLE,
       CONSTRAINT "JOB_MOBILE_CHK" CHECK (contacttelephone NOT LIKE '%[^0-9]%') ENABLE,
       CONSTRAINT "JOBS_SITE_ID_FK" FOREIGN KEY ("SITEID")
        REFERENCES "APEX811"."SITES" ("SITEID") ENABLE
   ) ;

CREATE SEQUENCE  "APEX811"."SEQ_JOB_ID"  MINVALUE 1 MAXVALUE 99999 INCREMENT BY 1 START WITH 10024
NOCACHE  NOORDER  CYCLE  NOKEEP  NOSCALE  GLOBAL ;

CREATE TABLE "APEX811"."PROGRAMMES"
   (  "PROGRAMME_CODE" CHAR(4 BYTE),
      "PROGRAMME_NAME" VARCHAR2(60 BYTE) CONSTRAINT "PROGRAME_NAME_NN" NOT NULL ENABLE,
      "PLACEMENT_MANDATORY" CHAR(1 BYTE),
       CONSTRAINT "PROGRAMME_CODE_PK" PRIMARY KEY ("PROGRAMME_CODE")
   ) ;
```

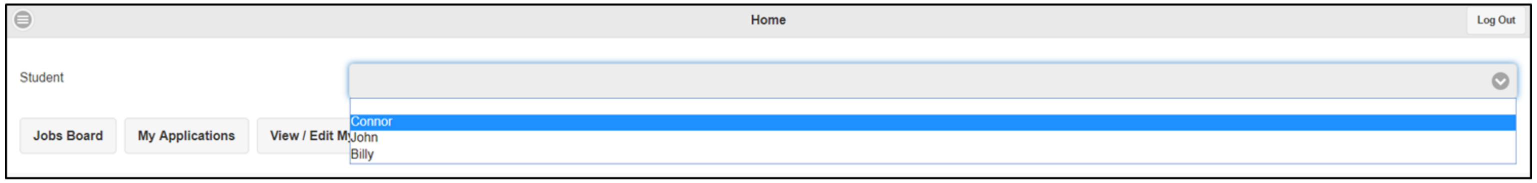
```
CREATE TABLE "APEX811"."STUDENTS"
   (  "STUDENTID" NUMBER DEFAULT ON NULL "APEX811"."SEQ_STUDENT_ID"."NEXTVAL" NOT NULL ENABLE,
      "USERNAME" VARCHAR2(20 BYTE) CONSTRAINT "USERNAME_NN" NOT NULL ENABLE,
      "PASSWORD" VARCHAR2(20 BYTE) CONSTRAINT "PASSWORD_NN" NOT NULL ENABLE,
      "FIRSTNAME" VARCHAR2(30 BYTE) CONSTRAINT "FIRSTNAME_NN" NOT NULL ENABLE,
      "LASTNAME" VARCHAR2(45 BYTE) CONSTRAINT "LASTNAME_NN" NOT NULL ENABLE,
      "EMAIL" VARCHAR2(100 BYTE) CONSTRAINT "STUDENT_EMAIL_NN" NOT NULL ENABLE,
      "MOBILE" VARCHAR2(11 BYTE),
      "LANDLINE" VARCHAR2(11 BYTE),
      "PROGRAMME_CODE" CHAR(4 BYTE) CONSTRAINT "PROGRAMME_CODE_NN" NOT NULL ENABLE,
      "DOB" DATE CONSTRAINT "DOB_NN" NOT NULL ENABLE,
      "PREFERENCE" VARCHAR2(2000 BYTE),
      "TERMADDRESS" VARCHAR2(200 BYTE) CONSTRAINT "STUDENT_TERMADDRESS_NN" NOT NULL ENABLE,
      "TERMPOSTCODE" VARCHAR2(8 BYTE) CONSTRAINT "STUDENT_TERMPOSTCODE_NN" NOT NULL ENABLE,
      "HOMEADDRESS" VARCHAR2(200 BYTE) CONSTRAINT "STUDENT_HOMEADDRESS_NN" NOT NULL ENABLE,
      "HOMEPOSTCODE" VARCHAR2(8 BYTE) CONSTRAINT "STUDENT_HOMEPOSTCODE_NN" NOT NULL ENABLE,
      "CVSUBMITTED" DATE,
      "CVACCEPTED" DATE,
       CONSTRAINT "STUDENTID_PK" PRIMARY KEY ("STUDENTID"),
       CONSTRAINT "STUDENT_EMAIL_CHK" CHECK (email LIKE  '%_@___%.__%') ENABLE,
       CONSTRAINT "STUDENT_MOBILE_CHK" CHECK (mobile NOT LIKE '%[^0-9]%') ENABLE,
       CONSTRAINT "STUDENT_LANDLINE_CHK" CHECK (landline NOT LIKE '%[^0-9]%') ENABLE,
       CONSTRAINT "STUDENT_TERMPOSTCODE_CHK" CHECK (termpostcode LIKE '%___ ___') ENABLE,
       CONSTRAINT "STUDENT_HOMEPOSTCODE_CHK" CHECK (homepostcode LIKE '%___ ___') ENABLE,
       CONSTRAINT "FK_PROGRAMME_CODE" FOREIGN KEY ("PROGRAMME_CODE")
        REFERENCES "APEX811"."PROGRAMMES" ("PROGRAMME_CODE") ENABLE
   ) ;

CREATE SEQUENCE  "APEX811"."SEQ_STUDENT_ID"  MINVALUE 1 MAXVALUE 999999 INCREMENT BY 1 START WITH 100042
NOCACHE  NOORDER  CYCLE  NOKEEP  NOSCALE  GLOBAL ;
```

```
CREATE OR REPLACE EDITIONABLE TRIGGER "APEX811"."TRG_CV_ACCEPTED"
BEFORE UPDATE OF CVACCEPTED ON STUDENTS FOR EACH ROW
BEGIN
IF :OLD.CVSUBMITTED IS NULL
THEN :NEW.CVACCEPTED := NULL;
END IF;
END;
ALTER TRIGGER "APEX811"."TRG_CV_ACCEPTED" ENABLE;

CREATE OR REPLACE EDITIONABLE TRIGGER "APEX811"."TRG_DOB"
BEFORE INSERT OR UPDATE OF DOB ON STUDENTS FOR EACH ROW
BEGIN
IF (:NEW.dob + NUMTOYMINTERVAL(18,'YEAR')) >
SYSDATE THEN
RAISE_APPLICATION_ERROR(-18, 'Student must be at least 18 years old');
END IF;
END;
ALTER TRIGGER "APEX811"."TRG_DOB" ENABLE;


CREATE TABLE "APEX811"."APPLICATIONS"
   (  "APPLICATIONID" VARCHAR2(9 BYTE) DEFAULT ON NULL
'A'||LTRIM(TO_CHAR("APEX811"."SEQ_APPLICATION_ID"."NEXTVAL",'0999999')) NOT NULL ENABLE,
      "STUDENTID" NUMBER CONSTRAINT "STUDENT_ID_NN" NOT NULL ENABLE,
      "JOBID" VARCHAR2(7 BYTE) CONSTRAINT "JOB_ID_NN" NOT NULL ENABLE,
      "APPLICATIONSTATUS" VARCHAR2(28 BYTE) CONSTRAINT "APPLICATIONSTATUS_NN" NOT NULL ENABLE,
      "STATUSDATE" DATE,
       CONSTRAINT "APPLICATION_ID_PK" PRIMARY KEY ("APPLICATIONID"),
       CONSTRAINT "APPLICATIONS_STUDENT_ID_FK" FOREIGN KEY ("STUDENTID")
        REFERENCES "APEX811"."STUDENTS" ("STUDENTID") ENABLE,
       CONSTRAINT "FK_JOBID" FOREIGN KEY ("JOBID")
        REFERENCES "APEX811"."JOBS" ("JOBID") ENABLE
   ) ;

CREATE SEQUENCE  "APEX811"."SEQ_APPLICATION_ID"  MINVALUE 1 MAXVALUE 9999999 INCREMENT BY 1 START WITH
1000029 NOCACHE  NOORDER  CYCLE  NOKEEP  NOSCALE  GLOBAL ;
```

```
CREATE OR REPLACE TRIGGER TRG_APPLICATION_UPDATE
BEFORE UPDATE OF APPLICATIONSTATUS, STATUSDATE ON APPLICATIONS FOR EACH ROW
DECLARE
        HISTORY_STATUS VARCHAR(40);
        HISTORY_STATUS_COUNT NUMBER;
BEGIN
    SELECT COUNT(APPLICATIONSTATUS) INTO HISTORY_STATUS_COUNT
    FROM APPLICATIONHISTORIES
    WHERE APPLICATIONHISTORIES.APPLICATIONID = :NEW.APPLICATIONID
    AND APPLICATIONHISTORIES.APPLICATIONSTATUS = :NEW.APPLICATIONSTATUS;

    IF HISTORY_STATUS_COUNT > 0 THEN
    SELECT APPLICATIONHISTORIES.APPLICATIONSTATUS INTO HISTORY_STATUS
    FROM APPLICATIONHISTORIES
    WHERE APPLICATIONHISTORIES.APPLICATIONID = :NEW.APPLICATIONID
    AND APPLICATIONHISTORIES.APPLICATIONSTATUS = :NEW.APPLICATIONSTATUS;
    ELSE
        HISTORY_STATUS := NULL;
    END IF;

    IF :OLD.APPLICATIONSTATUS = :NEW.APPLICATIONSTATUS OR HISTORY_STATUS = :NEW.APPLICATIONSTATUS OR
    THEN
        RAISE_APPLICATION_ERROR(-20007,'You cannot select the same stage twice');
    ELSE
        INSERT INTO APPLICATIONHISTORIES
            (
            APPLICATIONID, APPLICATIONSTATUS, STATUSDATE
            )
        VALUES
            (
            :OLD.APPLICATIONID, :OLD.APPLICATIONSTATUS, :OLD.STATUSDATE
            );
    END IF;
END;
```

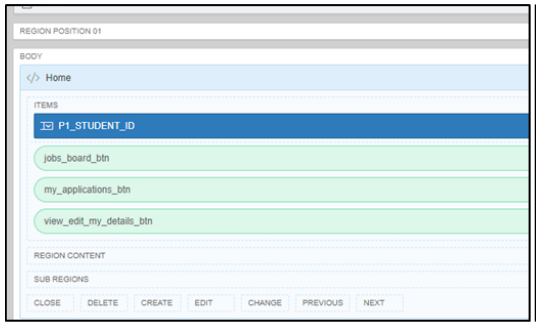
```
CREATE OR REPLACE TRIGGER CREATE_APPLICATION_TRIGGER
BEFORE INSERT OR UPDATE OF STUDENTID ON APPLICATIONS FOR EACH ROW
DECLARE ACCEPTEDCV DATE;
        ALREADYAPPLIEDNUMBER NUMBER;
BEGIN
    SELECT COUNT(APPLICATIONS.STUDENTID) INTO ALREADYAPPLIEDNUMBER
    FROM APPLICATIONS
    WHERE APPLICATIONS.STUDENTID = :NEW.STUDENTID AND APPLICATIONS.JOBID = :NEW.JOBID;

    IF ALREADYAPPLIEDNUMBER > 0
    THEN
        RAISE_APPLICATION_ERROR(-20008, 'YOU HAVE ALREADY APPLIED TO THIS JOB.');
    END IF;

    SELECT STUDENTS.CVACCEPTED INTO ACCEPTEDCV
    FROM STUDENTS
    WHERE STUDENTS.STUDENTID = :NEW.STUDENTID;
    IF ACCEPTEDCV IS NULL
    THEN
        RAISE_APPLICATION_ERROR(-20006, 'YOU MUST HAVE AN APPROVED CV.');
    END IF;
END;

CREATE TABLE "APEX811"."APPLICATIONHISTORIES"
   (  "APPLICATIONID" VARCHAR2(9 BYTE),
      "APPLICATIONSTATUS" VARCHAR2(28 BYTE) CONSTRAINT "APPLICATIONHISTORIESSTATUS_NN" NOT NULL ENABLE,
      "STATUSDATE" DATE,
       CONSTRAINT "APPLICATIONHISTORIES_PK" PRIMARY KEY ("APPLICATIONID", "APPLICATIONSTATUS"),
       CONSTRAINT "APPLICATION_HISTORIES_ID_FK" FOREIGN KEY ("APPLICATIONID")
        REFERENCES "APEX811"."APPLICATIONS" ("APPLICATIONID") ENABLE
   ) ;
```

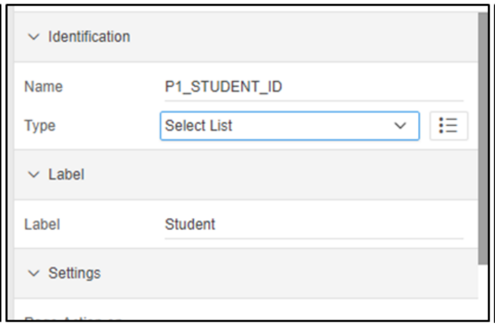
4. Student App Development Screenshots



Student

Connor
John
Billy





Identification

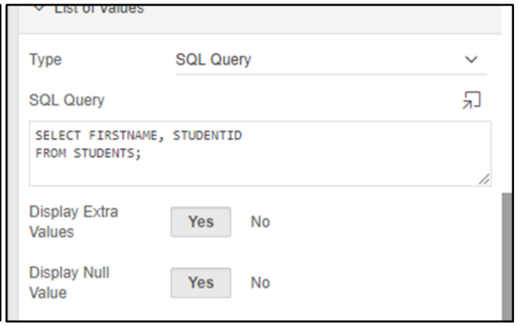
Name: P1_STUDENT_ID

Type: Select List

Label

Label: Student

Settings



List of Values

Type: SQL Query

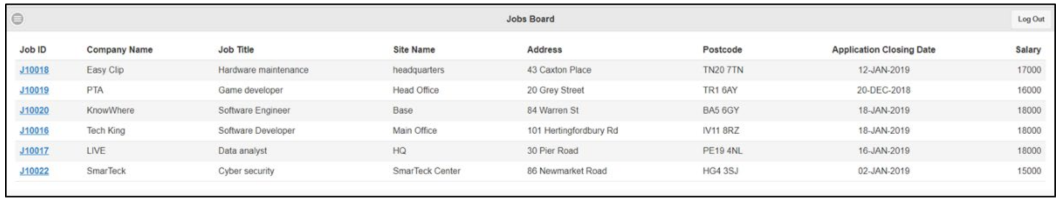
SQL Query

```
SELECT FIRSTNAME, STUDENTID
FROM STUDENTS;
```

Display Extra Values: Yes No

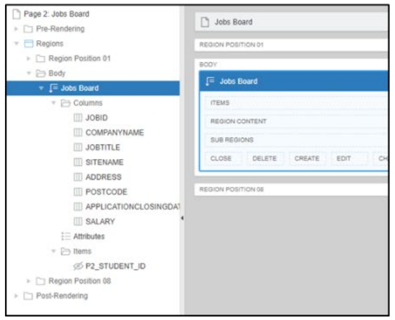
Display Null Value: Yes No

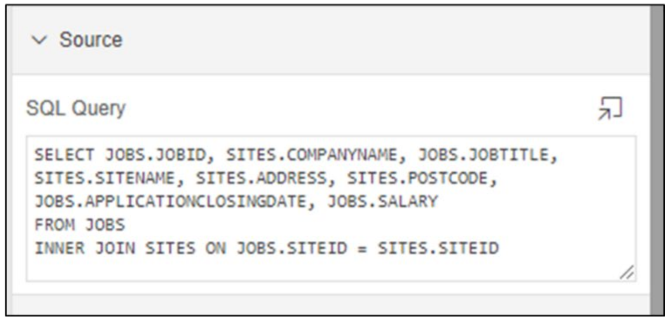
'Home' displays navigation buttons to take the user to other pages, which includes 'Jobs Boards', 'My Applications' and 'View/Edit My Details'. Also contains a Select List that displays all the students' names within the drop-down menu.



Job ID	Company Name	Job Title	Site Name	Address	Postcode	Application Closing Date	Salary
J10018	Easy Clip	Hardware maintenance	headquarters	43 Caxton Place	TN20 7TN	12-JAN-2019	17000
J10019	PTA	Game developer	Head Office	20 Grey Street	TR1 6AY	20-DEC-2018	16000
J10020	KnowWhere	Software Engineer	Base	84 Warren St	BA5 6GY	18-JAN-2019	18000
J10016	Tech King	Software Developer	Main Office	101 Hertingfordbury Rd	IV11 8RZ	18-JAN-2019	18000
J10017	LIVE	Data analyst	HQ	30 Pier Road	PE19 4NL	16-JAN-2019	18000
J10022	SmarTeck	Cyber security	SmarTeck Center	86 Newmarket Road	HG4 3SJ	02-JAN-2019	15000

'Jobs Board' contains a reflow report which uses a SQL query to display all relevant information of a job in a table. The user can click on the 'Job ID' to be taken to a separate page showing all information specific to the job. Useful for information that does not fit into the table, such as the job description.

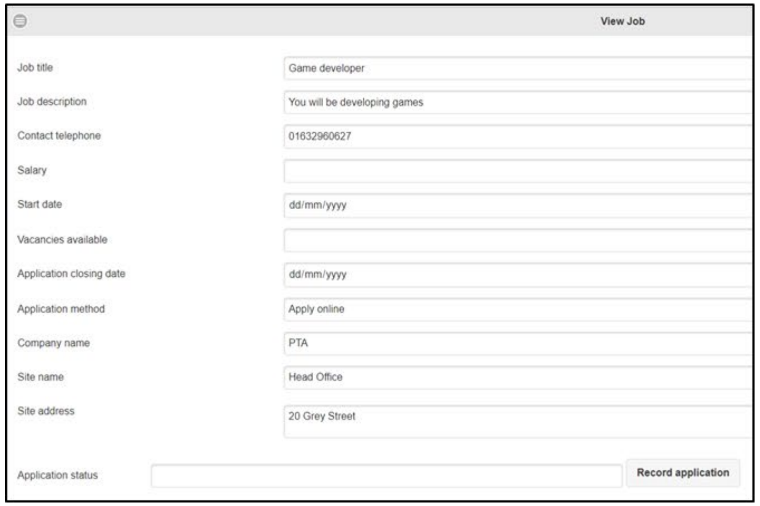




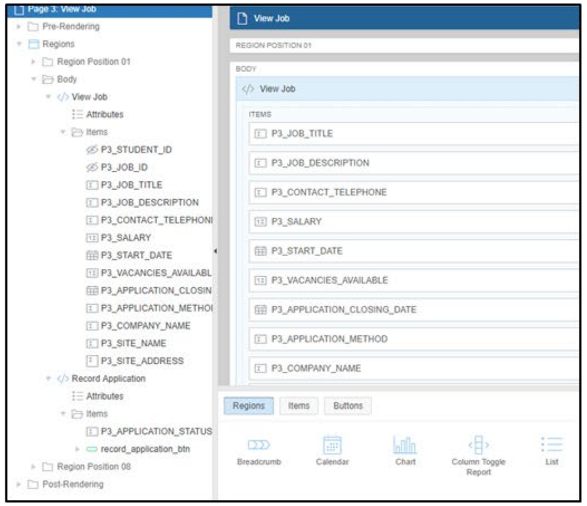
Source

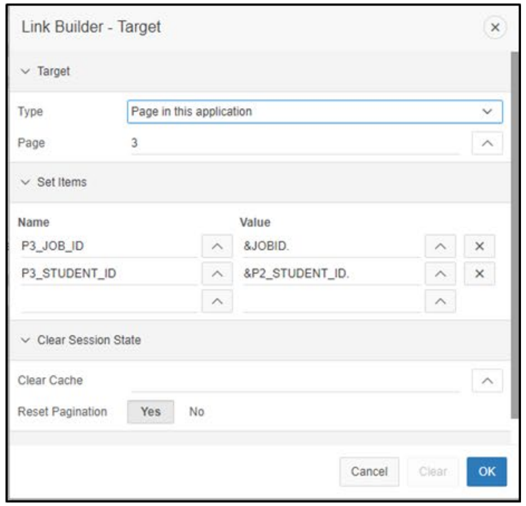
SQL Query

```
SELECT JOBS.JOBID, SITES.COMPANYNAME, JOBS.JOBTITLE,
SITES.SITENAME, SITES.ADDRESS, SITES.POSTCODE,
JOBS.APPLICATIONCLOSINGDATE, JOBS.SALARY
FROM JOBS
INNER JOIN SITES ON JOBS.SITEID = SITES.SITEID
```


View Job

Job title	Game developer
Job description	You will be developing games
Contact telephone	01632960627
Salary	
Start date	dd/mm/yyyy
Vacancies available	
Application closing date	dd/mm/yyyy
Application method	Apply online
Company name	PTA
Site name	Head Office
Site address	20 Grey Street
Application status	




Link Builder - Target

Target

Type: Page in this application

Page: 3

Set Items

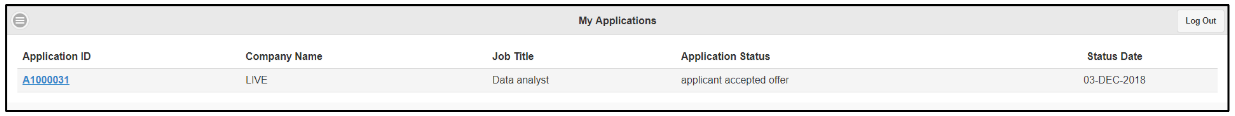
Name	Value
P3_JOB_ID	&JOBID.
P3_STUDENT_ID	&P2_STUDENT_ID.

Clear Session State

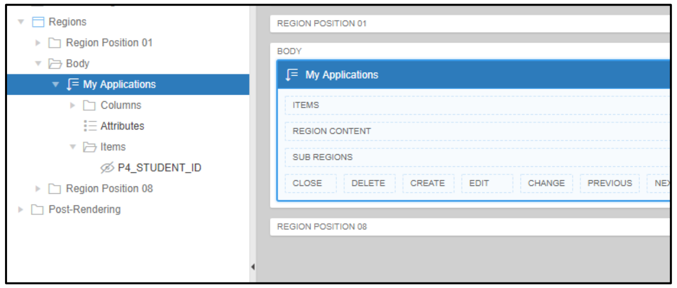
Clear Cache:

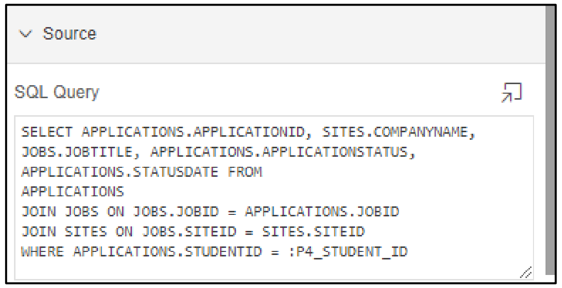
Reset Pagination: [✓ Yes] No

'View Job' contains Static Content with columns relevant to the Job's details. In order to transfer the information across from 'Jobs Board' to 'View Job', the settings of the column had to be 'Link', which will navigate the user to the 'View Job' page. The text fields must be automatically filled with the appropriate information when this happens, and therefore we were required to use the 'Set Name' property.


My Applications

Application ID	Company Name	Job Title	Application Status	Status Date
A1000031	LIVE	Data analyst	applicant accepted offer	03-DEC-2018

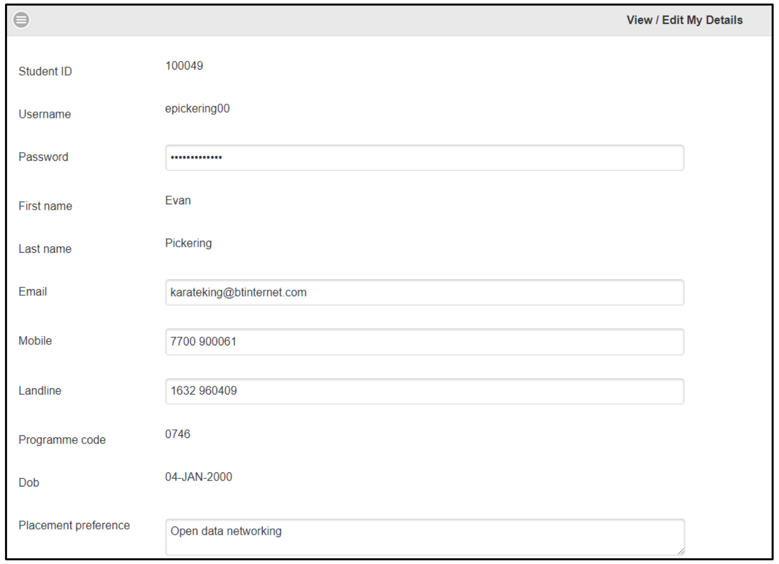



Source

SQL Query

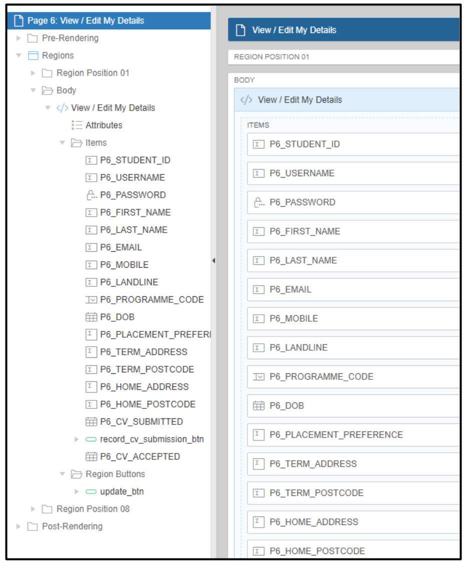
```
SELECT APPLICATIONS.APPLICATIONID, SITES.COMPANYNAME,
JOBS.JOBTITLE, APPLICATIONS.APPLICATIONSTATUS,
APPLICATIONS.STATUSDATE FROM
APPLICATIONS
JOIN JOBS ON JOBS.JOBID = APPLICATIONS.JOBID
JOIN SITES ON JOBS.SITEID = SITES.SITEID
WHERE APPLICATIONS.STUDENTID = :P4_STUDENT_ID
```

'My Applications' is very similar to the 'Jobs Board' page, in which it contains a reflow report which extracts specific information from the database and structures it in a table. The relevant headings are 'Application ID', 'Company Name', 'Job Title', 'Application Status' and 'Status Date'.



View / Edit My Details

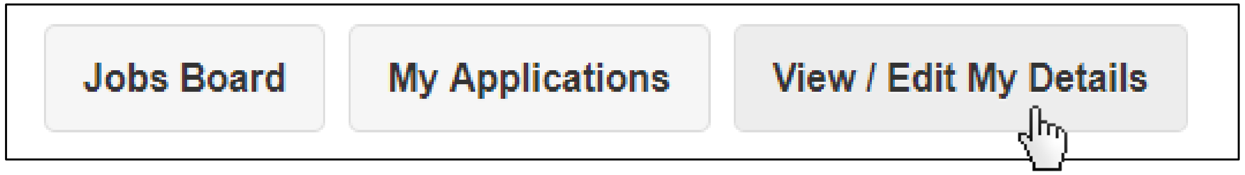
Student ID	100049
Username	epickering00
Password	•••••••••••••
First name	Evan
Last name	Pickering
Email	karateking@btinternet.com
Mobile	7700 900061
Landline	1632 960409
Programme code	0746
Dob	04-JAN-2000
Placement preference	Open data networking



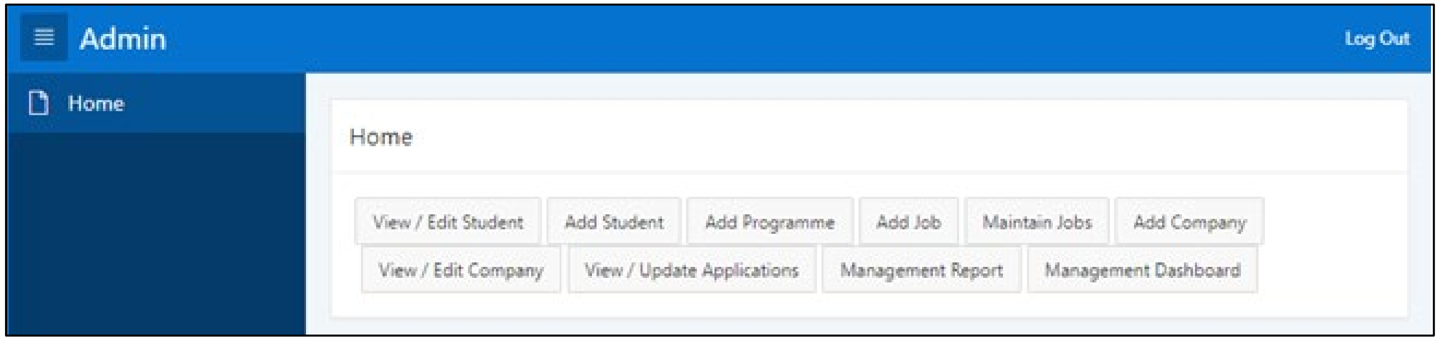
```
Page 6: View / Edit My Details
  Pre-Rendering
  Regions
    Region Position 01
    Body
      View / Edit My Details
        Attributes
        Items
          P6_STUDENT_ID
          P6_USERNAME
          P6_PASSWORD
          P6_FIRST_NAME
          P6_LAST_NAME
          P6_EMAIL
          P6_MOBILE
          P6_LANDLINE
          P6_PROGRAMME_CODE
          P6_DOB
          P6_PLACEMENT_PREFERI
          P6_TERM_ADDRESS
          P6_TERM_POSTCODE
          P6_HOME_ADDRESS
          P6_HOME_POSTCODE
          P6_CV_SUBMITTED
          record_cv_submission_btn
          P6_CV_ACCEPTED
      Region Buttons
        update_btn
    Region Position 08
  Post-Rendering
```

```
View / Edit My Details
REGION POSITION 01
BODY
View / Edit My Details
ITEMS
P6_STUDENT_ID
P6_USERNAME
P6_PASSWORD
P6_FIRST_NAME
P6_LAST_NAME
P6_EMAIL
P6_MOBILE
P6_LANDLINE
P6_PROGRAMME_CODE
P6_DOB
P6_PLACEMENT_PREFERENCE
P6_TERM_ADDRESS
P6_TERM_POSTCODE
P6_HOME_ADDRESS
P6_HOME_POSTCODE
```

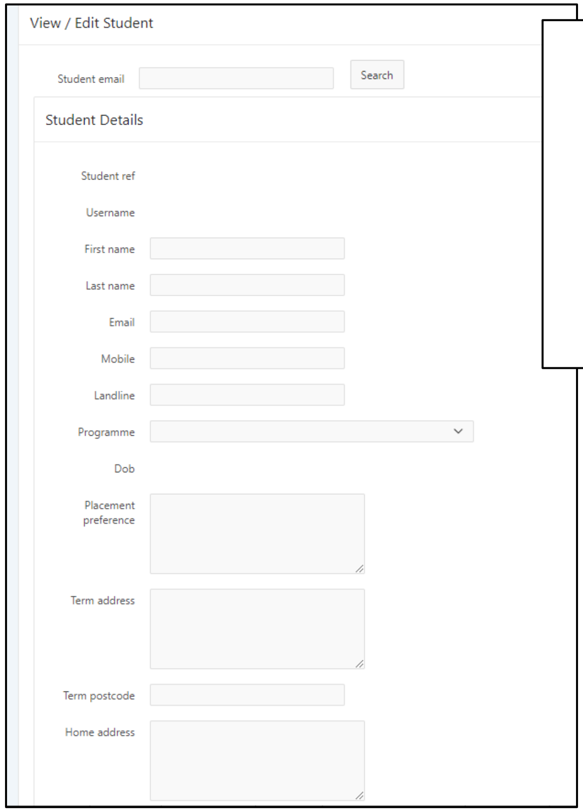
‘View/Edit My Details’ displays the user’s details; this is done by using similar methods for ‘View Job’ page.

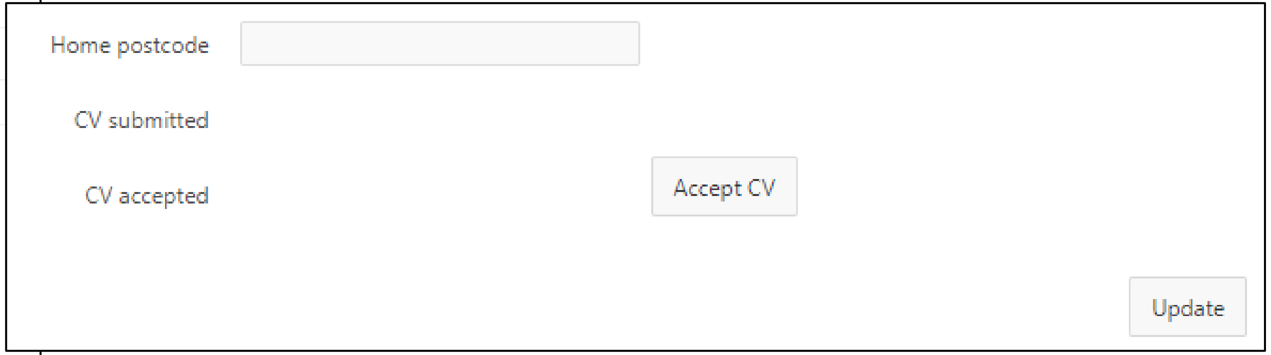


5. Admin App Development Screenshots



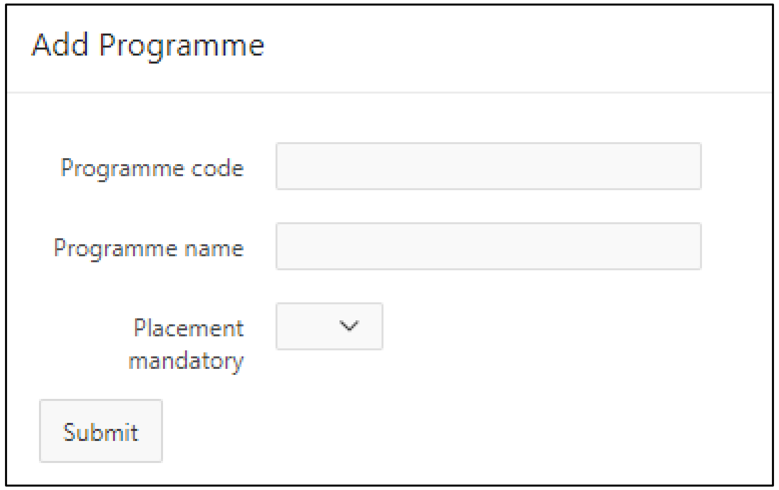
1 – Home page





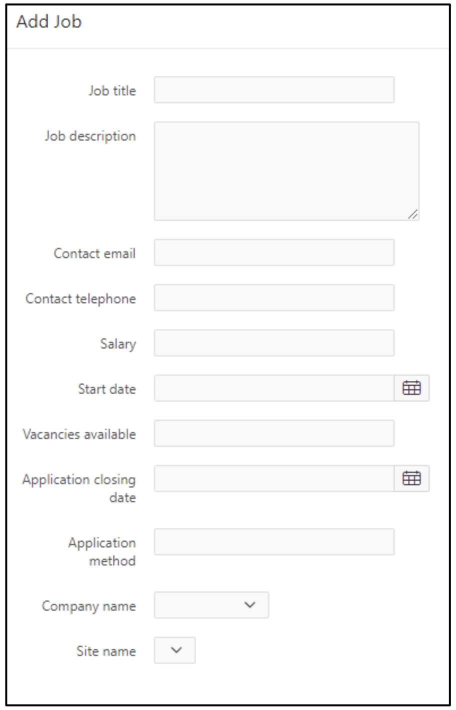
2 – View/Edit students

Specific pieces of data, such as student reference number and Date of Birth, are uneditable. This is to provide client side data integrity of vital information. When clicking the Accept CV Button, the current date is added as the CV accepted date if there is a CV submitted date.



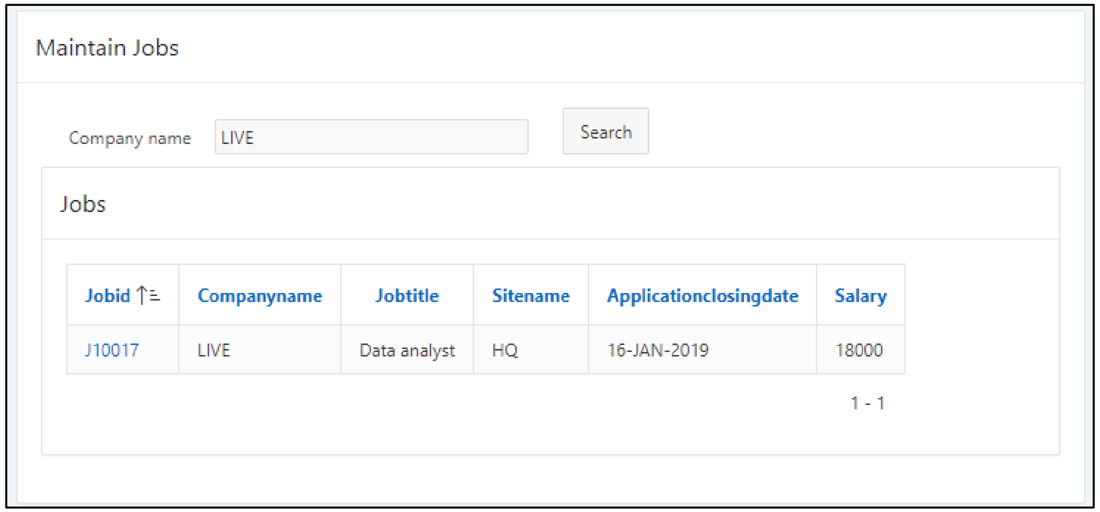
3 – Add programmes

Enables admin to add a new programme to the system.



4 – Add job

Enables admin to add a job to the system.

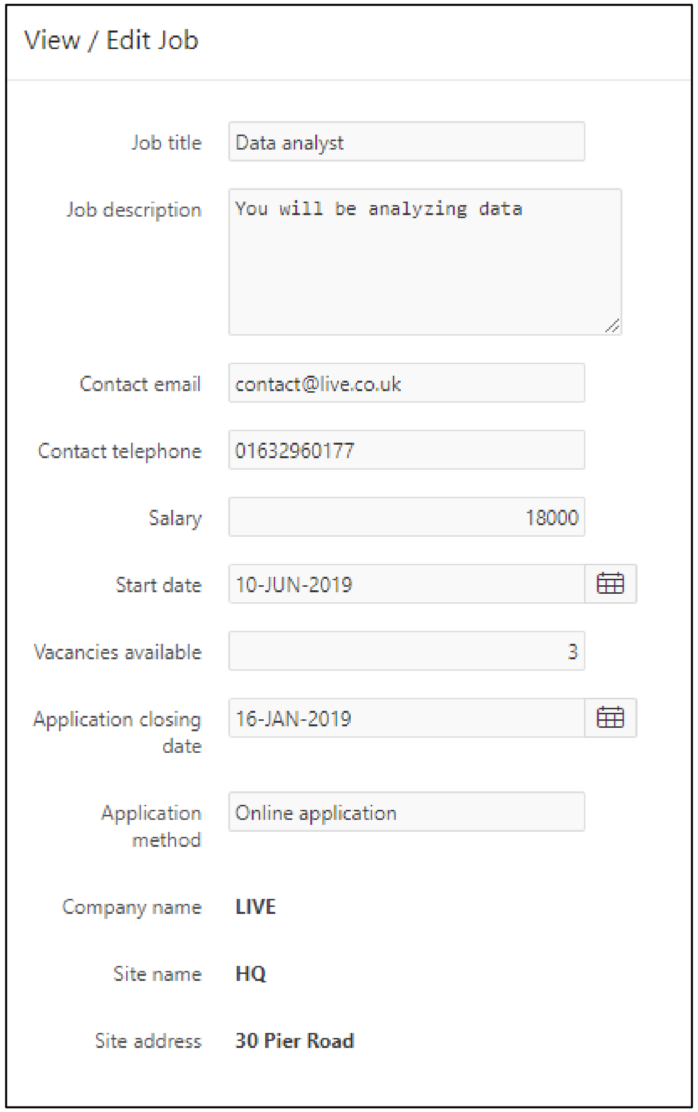

Maintain Jobs

Company name LIVE

Jobs

Jobid	Companyname	Jobtitle	Sitename	Applicationclosingdate	Salary
J10017	LIVE	Data analyst	HQ	16-JAN-2019	18000

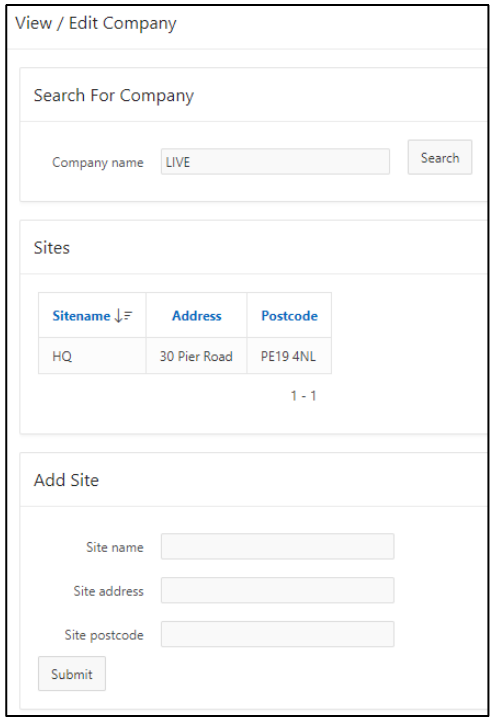
1 - 1


View / Edit Job

Job title	Data analyst
Job description	You will be analyzing data
Contact email	contact@live.co.uk
Contact telephone	01632960177
Salary	18000
Start date	10-JUN-2019
Vacancies available	3
Application closing date	16-JAN-2019
Application method	Online application
Company name	**LIVE**
Site name	**HQ**
Site address	**30 Pier Road**

5 – Manage jobs

Clicking on a job ID will take the user to another page where they are able to manage the jobs details.


View / Edit Company

Search For Company

Company name: LIVE

Search

Sites

Sitename	Address	Postcode
HQ	30 Pier Road	PE19 4NL

1 - 1

Add Site

Site name

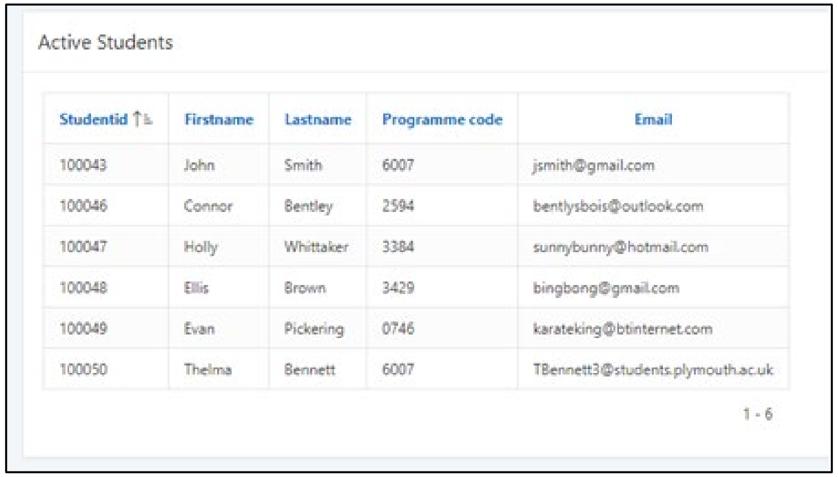
Site address

Site postcode

Submit

6 – View/edit company

To add a site, the user must first search for a company. The company searched for is then added to the site as a foreign key.

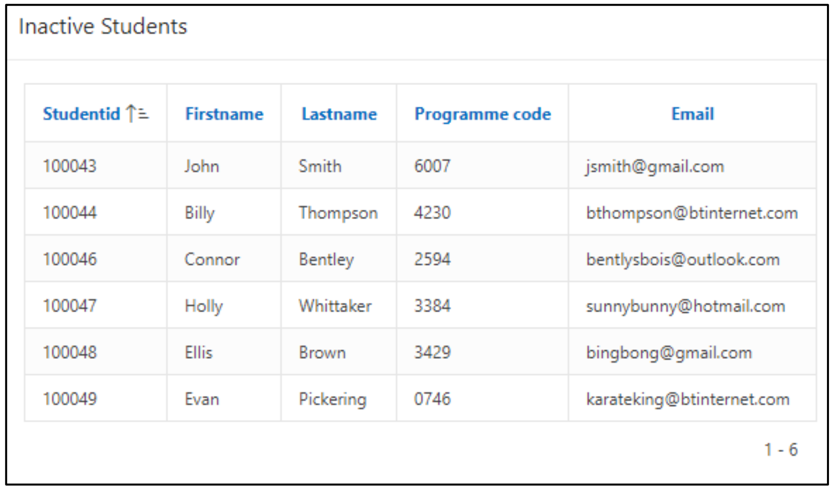

Active Students

Studentid	Firstname	Lastname	Programme code	Email
100043	John	Smith	6007	jsmith@gmail.com
100046	Connor	Bentley	2594	bentlysbois@outlook.com
100047	Holly	Whittaker	3384	sunnybunny@hotmail.com
100048	Ellis	Brown	3429	bingbong@gmail.com
100049	Evan	Pickering	0746	karateking@btinternet.com
100050	Thelma	Bennett	6007	TBennett3@students.plymouth.ac.uk

1 - 6

8.1 – Management reports: Active students

Shows a list of active students



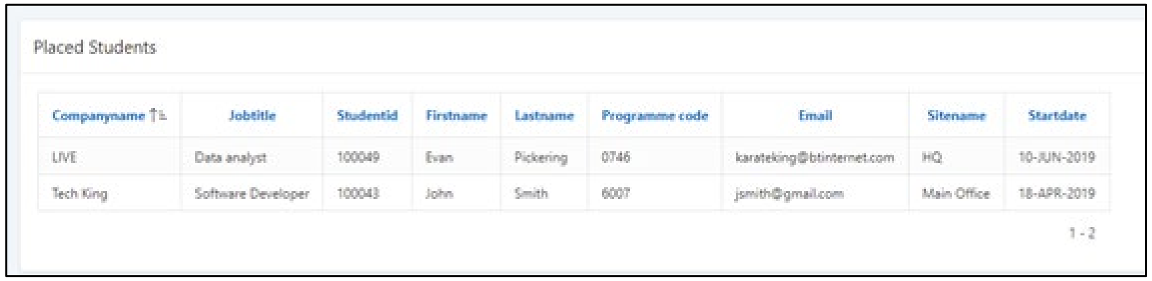
Inactive Students

Studentid	Firstname	Lastname	Programme code	Email
100043	John	Smith	6007	jsmith@gmail.com
100044	Billy	Thompson	4230	bthompson@btinternet.com
100046	Connor	Bentley	2594	bentlysbois@outlook.com
100047	Holly	Whittaker	3384	sunnybunny@hotmail.com
100048	Ellis	Brown	3429	bingbong@gmail.com
100049	Evan	Pickering	0746	karateking@btinternet.com

1 - 6

8.2 – Management reports: Inactive students

Shows a list of inactive students.



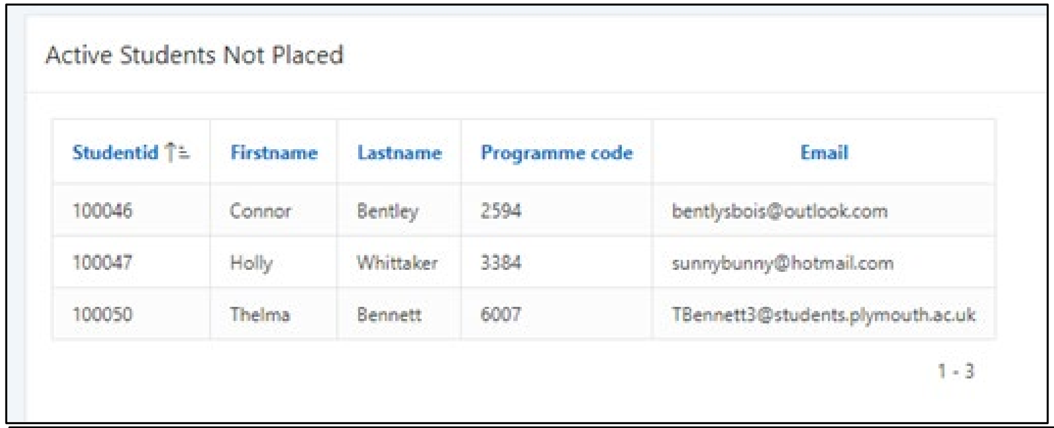
Placed Students

Companyname	Jobtitle	Studentid	Firstname	Lastname	Programme code	Email	Sitename	Startdate
LIVE	Data analyst	100049	Evan	Pickering	0746	karateking@btinternet.com	HQ	10-JUN-2019
Tech King	Software Developer	100043	John	Smith	6007	jsmith@gmail.com	Main Office	18-APR-2019

1 - 2

8.3 – Management reports: Placed students

Shows a list of placed students with the details of the job they are placed on.



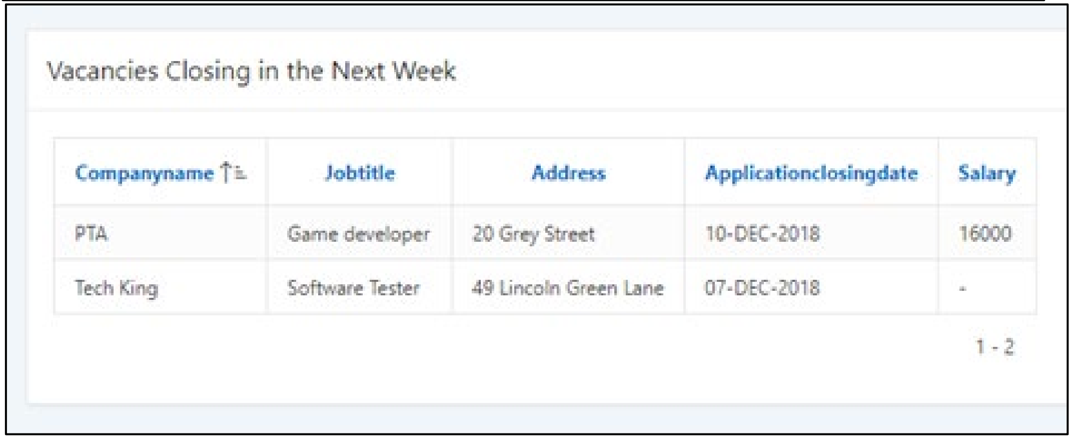
Active Students Not Placed

Studentid	Firstname	Lastname	Programme code	Email
100046	Connor	Bentley	2594	bentlysbois@outlook.com
100047	Holly	Whittaker	3384	sunnybunny@hotmail.com
100050	Thelma	Bennett	6007	TBennett3@students.plymouth.ac.uk

1 - 3

8.4 – Management reports: Active students not placed

Shows a list of active students that are not placed.



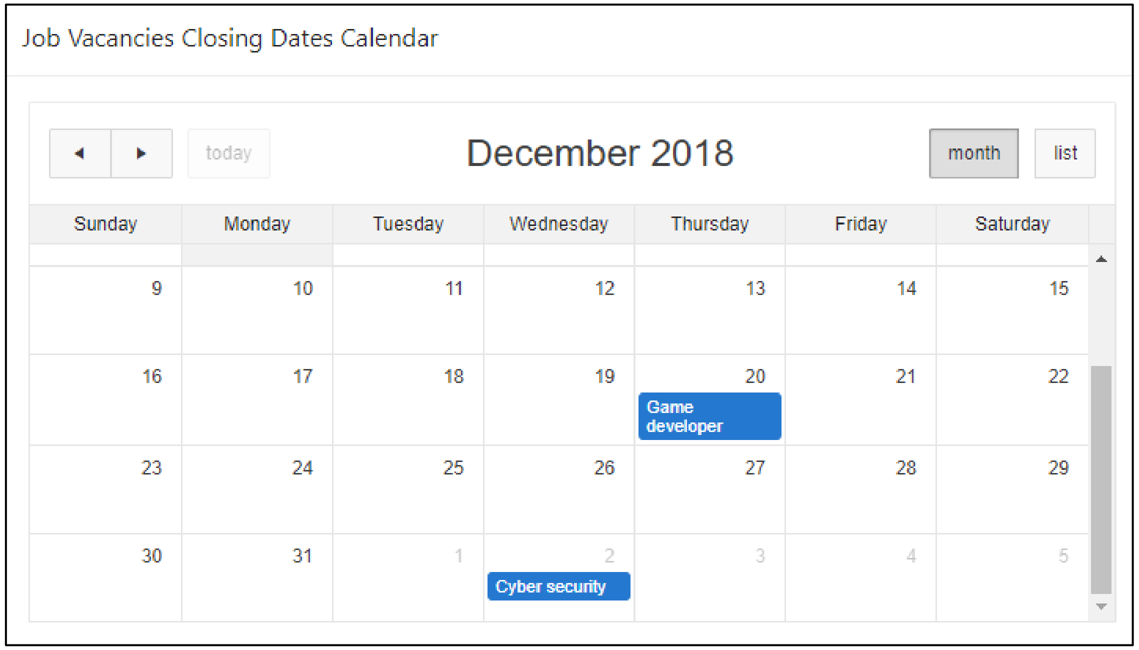
Vacancies Closing in the Next Week

Companyname	Jobtitle	Address	Applicationclosingdate	Salary
PTA	Game developer	20 Grey Street	10-DEC-2018	16000
Tech King	Software Tester	49 Lincoln Green Lane	07-DEC-2018	-

1 - 2

8.5 – Management reports: Closing vacancies

Show a list of job vacancies that are closing within the next 7 days.



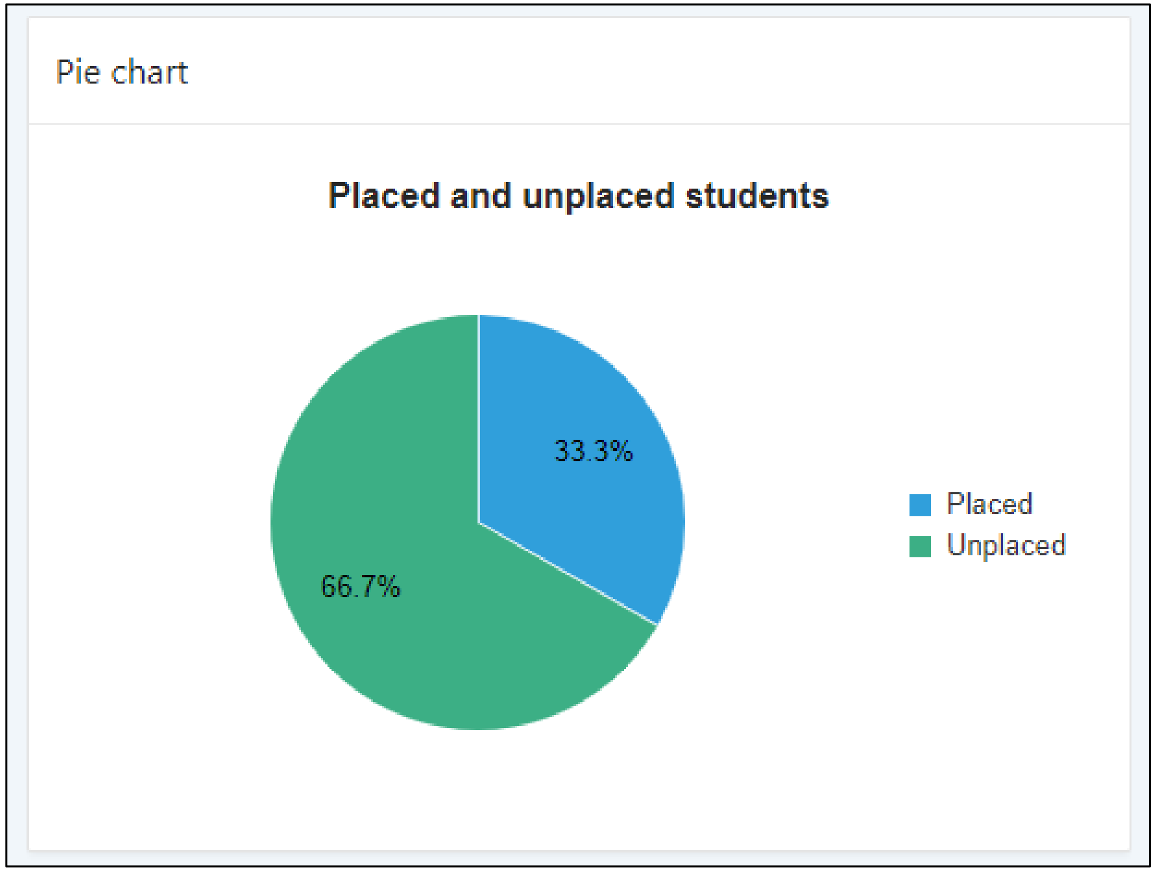
Job Vacancies Closing Dates Calendar

December 2018

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
9	10	11	12	13	14	15
16	17	18	19	20 Game developer	21	22
23	24	25	26	27	28	29
30	31	1	2 Cyber security	3	4	5

9 – Job vacancy calendar

Shows the closing dates of the job vacancies on a calendar with the ability to click on a job to take you to the job vacancy details.

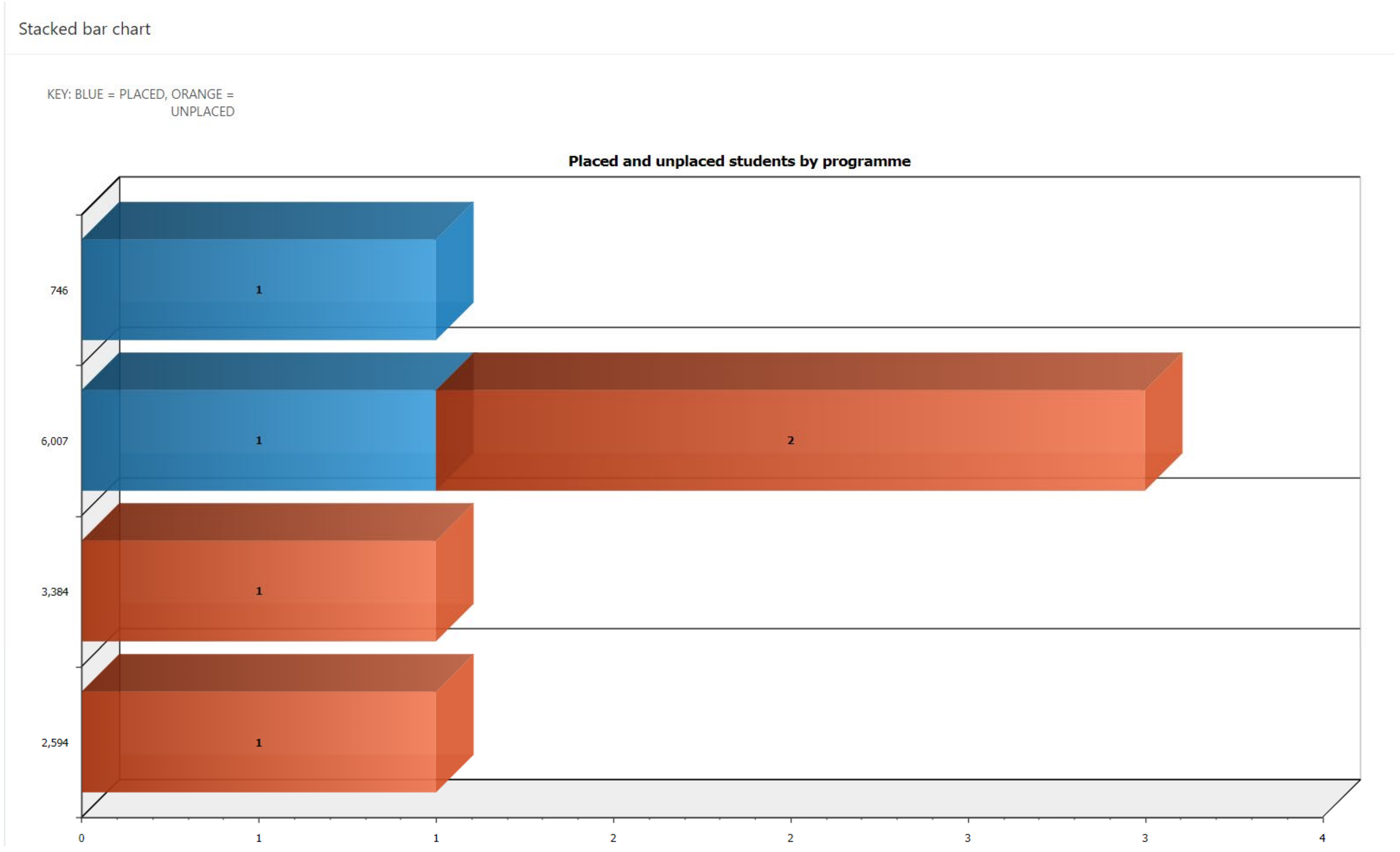


10 – Pie chart

Shows the proportion of place to unplaced active students.

11 – Stacked bar chart

A stacked bar chart to show the ratio of placed students in blue to unplaced students in orange, grouped by programme code.



Critical Appraisal

In this project, we have created an Oracle SQL Database for a Placement Application Tracking System. Furthermore, we have created both a Desktop Application that will enable the admin to carry out a number of tasks that including adding and maintain students, jobs and applications; and a Mobile application that will enable students to carry out tasks such as view and apply to jobs, view and update their application progress. However, this system does have a number of issues that could be improved in the future.

Firstly, this system has the ability to both add and update students from both the admin and the student application. This means that students and admins will be able to edit a student's information where the admin accounts have the extra ability to edit for example a student's name or date of birth should it of been entered incorrectly. Additionally, data integrity has been used where possible which is seen in the programme code section that provides the admin with a drop-down box to enable them to select which programme that student is on so the admin is not required to remember each individual programme code.

Another particularly successful element was the management reports which enable the admin to view tables of students that are active, inactive, placed, and active but not yet placed. Additional information about the student has been displayed such as their email should the admin want or need to contact them. Furthermore, in the management report, there is a closing vacancies section that shows a list of jobs that are closing within the next 7 days which would be useful as the admin could use this information to send out reminders to students.

Another worthy part of the system is the management dashboard. The dashboard includes a stacked bar chart that shows the number of placed and unplaced students of each programme; a pie chart that shows the percentage of the proportion of placed to unplaced active students and a calendar that shows the dates of when each job vacancy closes. Furthermore, on the dashboard calendar, the admin is able to click on each job which will produce a page showing that jobs details.

On the other hand, there are a number of pitfalls with this system. Firstly, the student web application does not implement student login accounts and currently, students are only able to access their accounts by clicking on their name in a drop-down menu which is not sustainable as it makes the system unprotected as any student can access another student's information. Additionally, passwords are also held in plain text inside the database and so makes this system very vulnerable to possible attacks. Therefore it is important that passwords should be encrypted when stored in the database.

Another weakness of the system is that the system only for example records the date on which a job application was submitted or updated and it would be more useful for the system to also record the time of which it was submitted.

Overall, it is clear that this system does meet the criteria set out for it in terms of providing all the basic functionality of maintaining the data within the database. Furthermore, it also provides management reports and a dashboard to enable the admin to view the data in a number of different ways. The key area, however, that does need to be rectified is the ability for students to log in with a username and password.