

COM1025 Web and Database Systems

Coursework Assignment

Surrey's Societies

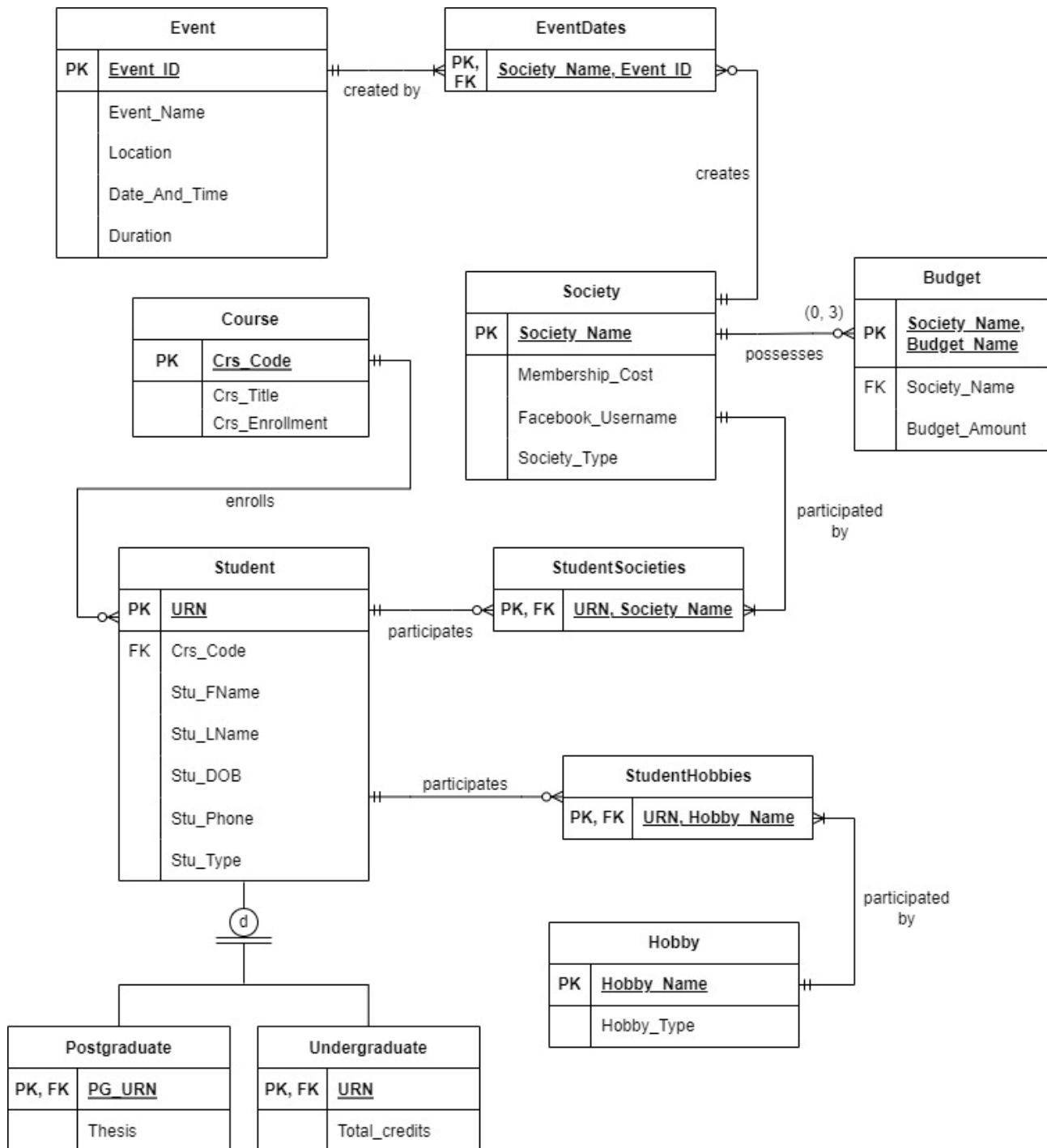
URN	680957
Username	Jc02842
Date	14/11/2023

1	Business Rules and Assumptions.....	1
2	Extended Entity Relationship Diagram (EERD)	2
3	Logical Relational Database Schema	3
4	Website Working with MySQL Database	4
5	Advanced Tasks	5
6	References	5
7	Appendix: Screenshots of Website.....	6

1 Business Rules and Assumptions

1. A university offers courses and has two categories of students: undergraduates and postgraduates.
 - a. The university needs to store the following details about each student: URN, course code, name, date of birth, a phone number they are on.
 - b. A student must be a postgraduate or an undergraduate, but they cannot be both at the same time.
 - c. A student must be enrolled on one and only one course, while a course can be enrolled by 0, 1 or many students.
 - d. A course has a course code, title and total student enrolment number
2. The university also wants to keep data on hobbies of students, if they wish to provide that information, so that appropriate clubs and societies can be suggested to them. The current list of hobby types include: water sports, sports, creative arts, performing arts, and fitness.
 - a. A student can have no hobby, one hobby or can have many hobbies.
 - b. A hobby can be related to no student or to many students.
 - c. The current list of hobbies include: reading, hiking, weightlifting, ballroom dancing, football, tennis, rugby, climbing, swimming, water polo and canoeing.
3. The university also offers societies to its students. The current societies offered are athletics, canoeing, computer science, drawing, football, running, and swimming.
 - a. Each society has a name, joining cost, Facebook username, type and budgets.
 - b. Each budget has a name and amount.
 - c. Each budget can only be associated with one society.
 - d. Each society can have up to 3 budgets.
 - e. A student can join 0, 1 or many societies.
 - f. A society can be joined by 0, 1 or many students.
4. Each society can have host events.
 - a. Each event must have an ID, location, date and time. It can also have a duration.
 - b. Each society can host 0, 1, or many events.
 - c. Each event can be hosted by 1, or many societies.

2 Extended Entity Relationship Diagram (EERD)



3 Logical Relational Database Schema

Course(Crs_Code, Crs_Title, Crs_enrollment)

PRIMARY KEY: Crs_Code

Student(URN, Crs_Code, Stu_FName, Stu_LName, Stu_DOB, Stu_Email, Stu_Type)

PRIMARY KEY: URN

FOREIGN KEY: Crs_Code REFERENCES **Course**(Crs_Code)

Postgraduate(URN, Thesis)

PRIMARY KEY: URN

FOREIGN KEY: URN REFERENCES **Student**(URN)

Undergraduate(URN, Total_credits)

PRIMARY KEY: URN

FOREIGN KEY: URN REFERENCES **Student**(URN)

StudentHobbies(URN, Hobby_Name)

PRIMARY KEY: URN, Hobby_Name

FOREIGN KEY: URN REFERENCES **Student**(URN), Hobby_Name REFERENCES **Hobby**(Hobby_Name)

Hobby(Hobby_Name, Hobby_Type)

PRIMARY KEY: Hobby_Name

StudentSocieties(URN, Society_Name)

PRIMARY KEY: URN, Society_Name

FOREIGN KEY: URN REFERENCES **Student**(URN), Society_Name REFERENCES **Society**(Society_Name)

Society(Society_Name, Membership_Cost, Facebook_Username, Society_Type)

PRIMARY KEY: Society_Name

Budget(Society_Name, Budget_Name, Budget_Amount)

PRIMARY KEY: Society_Name, Budget_Name

FOREIGN KEY: Society_Name REFERENCES **Society**(Society_Name)

EventDates(Society_Name, Event_ID)

PRIMARY KEY: Society_Name, Event_ID

FOREIGN KEY: Society_Name REFERENCES **Society**(Society_Name), Event_ID REFERENCES **Event**(Event_ID)

Event(Event_ID, Location, Date_And_Time, Duration)

PRIMARY KEY: Event_ID

4 Website Working with MySQL Database

node_modules Contains every installed dependency for the project.

main.css The style sheet for my website provided by Joe Appleton (see references).

Index.ejs The dashboard for my website, showing buttons for “Societies” and “Students”.

Societies.ejs Displays a table of all societies added to the database, including all attributes seen in the EERD. Includes the actions to edit or view that specific society.

Society-edit.ejs Allows the user to update the society’s cost and Facebook username. Validation is performed in both this file and in index.js.

Society-view.ejs Displays all attributes of the society, as well as its corresponding budgets and events. JavaScript is used to present a message if the society does not currently have any budgets/ events.

Student-edit.ejs Allows the student’s course and phone number to be updated. Validation on the course is done through a dropdown list of all courses available.

Student-new.ejs Accessed by pressing “New Student” at the bottom of the students table in **students.ejs** Allows the user to enter values for their: URN, name, DOB, course, phone and student type. Validation is performed on each field.

Student-view.ejs Displays all attributes of the students, as well as their hobbies, societies and events they are to attend for their societies they are a part of. Students are also recommended societies based on their hobbies. Please see section 5 for more details.

Students.ejs A list of students with their attributes, along with the ability to edit, view delete or add a new student.

.gitignore Tells git to ignore node_modules and package-lock.json.

Index.js Handles GET and POST requests for each web page of the website. It is the entry point for the website and will display the dashboard (index.ejs) upon entry. Uses various SQL commands to pass objects into the web pages to give them the ability to dynamically update.

Package-lock.json Holds information on the dependencies or packages installed for a node.js project, including their exact version numbers.

Package.json Contains descriptive and functional metadata about a project, such as a name, version, and dependencies. The file provides the npm package manager with various information to help identify the project and handle dependencies.

```
✓ website
  > node_modules
  ✓ public
    # main.css
  ✓ views
    <> index.ejs
    <> societies.ejs
    <> society-edit.ejs
    <> society-view.ejs
    <> student-edit.ejs
    <> student-new.ejs
    <> student-view.ejs
    <> students.ejs
  .gitignore
  JS index.js
  {} package-lock.json
  {} package.json
```

5 Advanced Tasks

Database Back-End Side

- Added 7 societies and went into details, such as events and a budget, on each.
- Implemented a total of 6 of my own tables, including many to many and one to many translations.
- Added ability for each society to hold events.
- Each society has up to three budgets. I found this information by confirming it with my canoeing club's treasurer.
- Normalised all relations to 3NF.
- Added cascading deletions. For example, if a student is deleted then information on their hobbies, in 'StudentHobbies', is also deleted.
- Many more, complex, SQL queries. Please see SQL_queries.

Web Front-End Side

- Total of 8 user-facing routes
- Ability to add students with HTML and JavaScript validation, with a corresponding message displayed (e.g. "Student not updated due to invalid phone number").
- Ability to delete students.
- When viewing a student, they are given a list of events that their societies are hosting, an itinerary as such.
- When viewing a society, the list of their events are shown, along with their budgets. Added JavaScript to display a corresponding message if there are no budgets or events appended onto that society.
- Added functionality to recommend societies. This was done by appending each hobby and society a type. For example, if the student had a hobby of swimming, that would come under the type of 'Water Sports'. This would mean the student would then get recommended societies that also had the type of 'water sports', such as canoeing. Other types I included are "Sports", "Creative Arts", "Performing Arts" and "Fitness". Included JavaScript to stop societies the student already participated in being recommended.

6 References

Geshe (2016) How to center a <h2>? Available at:

<https://stackoverflow.com/questions/41439730/how-to-center-a-h2>

Joe Appleton (2023) web-exercise-starting-files [source code]

<https://github.com/joeappleton18/web-exercise-starting-files> (Accessed 10/12/2023)

Joe Appleton (2023) long_tutorial. Available at:

<https://surrey.cloud.panopto.eu/Panopto/Pages/Viewer.aspx?id=1ef211d8-aa2c-43b6-ae8-b0cb00ea80a9> (Accessed between 10/12/2023 and 13/12/2023).

7 Appendix: Screenshots of Website

Societies.ejs

Index.ejs

University of Surrey's Students and Societies

Societies

Students

[Dash](#) / [Societies](#) / [View](#)

Athletics

Society-view.ejs

[Dash](#) / [Societies](#) / [Edit](#)

Edit Athletics

Society-edit.ejs

Update Cost (nearest £): 25

Update Facebook Username:

Update

[Dash](#) / [Students](#) / [View](#)

Students.ejs

[Dash](#) / [Students](#)

Students

Student ID	First Name	Last Name	DOB	Phone	Course Code	Type	Actions
612345	Sara	Khan	20/06/2002	01483112233	100	UG	Edit View Delete
612348	Iyabo	Ogunsola	21/04/2002	01483445566	100	UG	Edit View Delete
612349	Omar	Sharif	29/12/2001	01483778899	100	UG	Edit View Delete
612350	Yunli	Guo	07/06/2002	01483123456	100	UG	Edit View Delete
612351	Costas	Spiliotis	02/07/2002	01483234567	100	UG	Edit View Delete
612352	Tom	Jones	24/10/2001	01483456789	101	UG	Edit View Delete
612353	Simon	Larson	23/08/2002	01483998877	101	UG	Edit View Delete
612354	Sue	Smith	16/05/2002	01483776655	101	UG	Edit View Delete
New Student							

- URN: 612348
- DOB: 21/04/2002
- Course: BSc Computer Science (Code: 100, 150 enrolled)
- Phone: 01483445566
- Type: UG
- Hobbies
 - 1. Hiking
 - 2. Tennis
- Societies
 - 1. Football
- Recommended Societies
 - 1. Athletics
 - 2. Running

Iyabo Ogunsola

Student-view.ejs

Events to Attend

Club	Event Name	Location	Start Date and Time	Duration (mins)
Football	Weekly Pub Meetup	Aldershot Rd, Guildford GU3 3AA	Tue Nov 14 2023 18:30:00 GMT+0000 (Greenwich Mean Time)	
Football	New Years Meetup	9 Old Palace Rd, Guildford GU1 9 7TJ	Mon Jan 01 2024 00:00:00 GMT+0000 (Greenwich Mean Time)	

[Dash](#) / [Students](#) / Create a New Student

Create a New Student

Student-edit.ejs

[Dash](#) / [Students](#) / [Edit](#)

Edit Iyabo Ogunsola

Update Course: BSc Computer Science

Update Phone: 01483445566

Update

- URN:

- First Name:

- Last Name:

- **DOB:**

- Course:

- Phone:

- Student Type:

Under Graduate ▾

Submit

Student-new.ejs