|  |  |
| --- | --- |
| **Section 1 – General Assessment Information** | |
| **Decision Making Rules** | Every task must be completed satisfactorily to be assessed as competent in the unit.  *\* For graded units, competence must be demonstrated before a mark can be given.* |
| **Reasonable Adjustment** | Students may request reasonable adjustment for assessment tasks.  Reasonable adjustment usually involves varying:   * the processes for conducting the assessment (eg: allowing additional time, varying the venue) * the evidence gathering techniques (eg: oral rather than written questioning, use of a scribe, modifications to equipment)   However, the evidence collected must allow the student to demonstrate all requirements of the unit. |
| **Special Consideration** | Students can apply for Special Consideration where personal circumstances have adversely affected their task result or ability to undertake an assessment. A Special Consideration form can be completed prior to, but no later than 3 days after, the date of the assessment and submitted to the relevant Manager. |
| **Re-submission** *(where tasks are not satisfactorily completed)* | Assessment tasks that are not satisfactory can be resubmitted up until the end of the unit as scheduled on the Unit Outline. The timing on this may depend on the equipment required for this assessment task.  **NOTE**: Assessment tasks submitted for the first time after the end of the unit as scheduled on the Unit Outline will not be assessed and student should be told to re-enrol in the unit. |
| **Plagiarism** | There are serious penalties for plagiarism. Students must ensure that all assessments are their own work (or group work).  Please refer to <https://www.swinburne.edu.au/current-students/manage-course/exams-results-assessment/plagiarism-academic-integrity/> |

| **Section 2 – Student and Assessor Instructions** | |
| --- | --- |
| **Conditions** | *Must be completed in class within a given three hour time limit*  *Enterprise Architect must be used to create diagrams*  *Keep this document open as the assessor will provide feedback on the open document. Changes may be made to the submission based on feedback.*  *When the assessor has completed this assessment they will sign off with a code unique to you, do not leave until this code is entered in the assessor signature section.* |
| **Task Overview and/or Description** | **Computers, Rooms and Classes**  Dod&Gy have opened up a new venture into delivering short courses. They have access to a few rooms in a building in the CBD. Some of the rooms have been outfitted with computers. The data for their setup at the moment can be accessed from the below connection string:  "Server=tcp:civapi.database.windows.net (Links to an external site.),1433;Initial Catalog=civapi;User ID=civ\_user;Password=Monday1330;"  ***The structure of the database is as follows:***  **Borrower**(id, surname, firstname, DOB)  **Author**(id, surname, firstname)  **Books**(ISBN, title, borrower, author)  FK (borrower) references (Borrower)  FK (author) references (Author)  ***The database tables uses the following datatypes:***  **Borrower**  id: int  surname: varchar  firstname: varchar  DOB: varchar  **Author**  id: int  surname: varchar  firstname: varchar  OO Programming Final Challenge - Attempt 1  **Books**  ISBN: int  title: varchar  borrower: int  author: int  They require a Web API that can deliver the following as JSON:   |  |  | | --- | --- | | **Endpoint** | **Description** | | /api/Borrowers | All Borrowers | | /api/Books | All Books (isbn, title) | | /api/Borrowers/{id} | Borrower with id = {id} & all book they have borrowed | | /api/Books/NotBorrowed | All Books that have not been borrowed (isbn, title) | | /api/Books/Borrowed | All Borrowed books including their borrower object | |
| **How the Assessment will be Conducted** | *Must be completed in class within a given three hour time limit*  *Assessment and feedback will be given immediately upon completion.* |
| **Submission Details** | 1. Link to GitHub repo 2. Marking Guide - below |

| **Section 3 – Assessment Criteria (Evidence to be Provided by the Student)** |
| --- |
| All of the required evidence within the task listed below must be satisfactorily demonstrated for the task to be assessed as satisfactory.  \* For graded units, the task must be satisfactorily completed before marks will be allocated. |

***Notes for the teacher***

*List in the Required Evidence column below all aspects of the task that are required to be demonstrated by the student for satisfactory completion of the task.*

*Name: Simon Holt*

*Student ID: 102570136*

*Date: 4/12/2019*

*Time: 10:58*

| **Marking Guide** | | | | |
| --- | --- | --- | --- | --- |
| **Required Evidence** | | **Satisfactory** | **Not Submitted** | **Unsatisfactory** |
| **1** | All Borrowers |  |  |  |
| **2** | All Books |  |  |  |
| **3** | Borrower (and Borrowed Books) By ID |  |  |  |
| **4** | Not Borrowed Books |  |  |  |
| **5** | Borrowed Books (and Borrower) |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4. Q&A** | | | | |
| **Required Evidence** | | | **Satisfactory** | **Not Submitted** | **Unsatisfactory** | |
| **1** | | How has the program been tested? How do you know if the JSON returned is accurate? |  |  |  | |
|  | | I used postman to get the data. And the data matched the needed descriptions for the endpoints |  |  |  | |
| **2** | | Give a brief explanation as to how a REST API works (~30 words) |  |  |  | |
|  | | A rest api takes the endpoints and uses them to find objects in a database and transfer them back in the form of JSON text. |  |  |  | |
| **3** | | How have objects been used in your program? |  |  |  | |
|  | | Yes i used them to return the data |  |  |  | |
| **4** | | Is a WEB API stateless? Is your API secure? |  |  |  | |
|  | | Yes they are. It is not secure in order to secure it requests would need authorization and data transfer would need to be encrypted. |  |  |  | |
| **5** | | Could the API have been built from another tech stack? Name at least one |  |  |  | |
|  | | Yes it could have. It could be built in node.js |  |  |  | |
| **6** | | Identify one site that could be used to find information on new web trends |  |  |  | |
|  | | <https://www.smashingmagazine.com/2019/03/web-design-trends-betheme/>  smashing magazine |  |  |  | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feedback to Student** | | | | |
| Click or tap here to enter text. | | | | |
| **Task Result** |  | **Satisfactory** |  | **Unsatisfactory** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Name** | **Signature** | **Date** |
| **Assessor** | Tim Baird | **1453056** | 4/12/2019 |