|  |  |  |
| --- | --- | --- |
| **School of Computing, Electrical and Applied Technology** | **ISCG 7424**  **Mobile Software Development**  **Semester 1, 2020** | **1** |
| **Due Date: 23:59:59 PM 04/April/2020** | |
| **Assignment One**  **Drivers Licence Test Booking System** | **Total Marks: 100**  **Weighting of this assignment: 30%** | |

**Assignment 1 – Individual**

Learning Outcomes

* Acquire in depth knowledge of a range of advanced technical features that extend the standard software development environment to cater for the development of software for mobile devices by obtaining information from the internet, manuals, textbooks and supplied sample code.
* Demonstrate the ability to successfully apply such features and techniques when writing code to solve selected problems in the given language for a mobile device.
* Communicate the knowledge from learning outcome 1, in a form that other programmers will find usable, relevant and easily intelligible.

Drivers Licence Test Booking System (DTBS)

Background: The NZTA wants to develop a *prototype* driver’s licence booking application suitable for Android devices.

Assume the following requirements:

* Booking Time slots start at 9:00
* Booking Time slots finish at 17:00 and the last booking slot is 16:00
* Each time slot is 1 hour long
* Bookings can be made from Monday to Friday and no bookings can be made on the weekend
* There are 8 booking slots per day
* There are a total of 10 driving instructors available for time slot
* The prototype only needs to allow bookings for one week

Scope Exclusions:

* record / manage the results of the driver’s licence test
* connect your application to a remote server / database to manage the timeslots

Requirements:

* Develop a series of wireframe screen designs using a screen design program of your choice to handle the requirements above. Make sure that the user will be able to navigate between each screen using an appropriate control / workflow.

[5 marks]-> //TODO

* Book a time slot for a given licence holder — bookTimeslot(String licenceNumber, String day, int hour)
* Return true if the slot could be booked successfully OR false if the slot cannot be booked
* A timeslot can be booked by 10 users at the same time
* One user can only book one time slot per day

[20 marks](DONE)

* Retrieve all booked time slots for a given licence holder — getTimeslotBooking(String licenceNumber)
* Return a sequence of timeslots

[20 marks](DONE)

* Get a list of summary data for timeslot bookings on a given day — getSlots(String day)
* Return a sequence of hours, bookingCounts ordered by hour

[20 marks](DONE)

* Create a mobile user interface to allow an end user to book a timeslot using appropriate Android widgets and layouts

[10 marks](DONE)

* Create a mobile user interface using a matrix structure to show bookings over a week colour coding booking hotspots

[10 marks](DONE)

* Create a suite of unit tests both positive and negative to exercise the core business functionality outlined in Steps 1-3.

[5 marks]

* Present working application to class and explain how the application works. Expect to be asked to make changes on the fly to your application and unit tests.

[5 marks]

Hints:

* Use in memory data structures such as Arrays, Collections, Maps etc.

Marking Schedule

|  |  |  |
| --- | --- | --- |
| **Task** | **Marking Criteria** | **Marks** |
| Screen Designs | Screen(s) designed with requirements in mind  Screens are aesthetically pleasing and allow for an intuitive and easy to follow form. | 4/5 |
| Screen Design Presentation | Screen designs presented and discussed with class | 0/5 |
| Book Slot | Timeslots run from 9:00 – 17:00 hours | 6/6 |
| Book Slot | Last bookable timeslot is 16:00 hours | 2/2 |
| Book Slot | Each time slot is 1 hour long | 2/2 |
| Book Slot | Bookings can be made from Monday to Friday and no bookings can be made on the weekend | 5/5 |
| Book Slot | There are 8 booking slots per day  There are a total of 10 driving instructors available for time slot  The prototype only needs to allow bookings for one week | 5/5 |
| Get Timeslot Booking | Code to retrieve timeslots by licence number | 10/10 check |
| Get Timeslot Booking | Code to retrieve sequence of timeslots | 10/10 ? because in arraylist they are |
| Summary Data | Code to retrieve list of timeslots for a given day | 10 |
| Summary Data | Code to generate a histogram / summary count of the booking total per day | 5 |
| Summary Data | Code to order summary data by hour | 5 |
| Booking Form | Appropriate layout + code to handle booking a timeslot for a given licence number | 10 |
| Summary Form | Matrix showing summary histogram data | 5 |
| Summary Form | Colour coding of matrix elements to show hotspots | 5 |
| Unit tests | Working units | 5 |
| Class presentation | Class presentation ability to change code on the fly | 5 |
| **Total Marks** | | **100** |

Late Submission of Assignments

Assignments submitted after the due date and time without having received an extension through Affected Performance Consideration (APC) will be penalised according to the following:

* 10% of marks deducted if submitted within 24hrs of the deadline,
* 20% of marks deducted if submitted after 24hrs and up to 48hrs of the deadline,
* 30% of marks deducted if submitted after 48hrs and up to 72hrs of the deadline,
* No grade will be given for an assignment that is submitted later than 72hrs after the deadline.

Special Assessment Circumstances

A student, who due to circumstances beyond his or her control, misses a test, final exam or an assignment deadline or considers his or her performance in a test, final exam or an assignment to have been adversely affected, should complete the Affected Performance Consideration (APC) form available from the Student Central.

When requesting an APC for an assignment, the APC must be submitted (along with work completed to date) within the time frame of the extension requested; i.e. if the Doctor’s certificate is for one (1) day, then the APC and work completed must be submitted within one (1) day.

Assistance to other Students

Students themselves can be an excellent resource to assist the learning of fellow students, but there are issues that arise in assessments that relate to the type and amount of assistance given by students to other students. It is important to recognize what types of assistance are beneficial to another’s learning and also what types of assistance are unacceptable in an assessment.

Beneficial Assistance

* Study Groups
* Discussion
* Sharing Reading Material
* Reading the available online and library resources

Unacceptable Assistance

* Working together on one copy of the assessment and submitting it as own work
* Giving another student your work
* Copying someone else’s work, this includes work done by someone not on the course
* Changing or correcting another student’s work
* Copying from books, the Internet etc. and submitting it as own work; anything taken directly from another source must be acknowledged correctly; show the source alongside the quotation