# Solent University

# Coursework Assessment Brief

# Assessment Details

|  |  |
| --- | --- |
| Unit Title: | Mobile Application Development |
| Unit Code: | SWD504 |
| Unit Leader: | Nick Whitelegg |
| Level: | 5 |
| Assessment Title: | PlacesToStay |
| Assessment Number: | AE1 |
| Assessment Type: | Software Development with report |
| Restrictions on Time/Word Count: | 2000-3000 words (guidance only) |
| Consequence of not meeting time/word count limit: | None; however it should not be necessary to exceed the word count above |
| Individual/Group: | Individual |
| Assessment Weighting: | 100% |
| Issue Date: | 20/1/20 |
| Hand In Date: | 1/5/20 |
| Planned Feedback Date: | 31/5/20 |
| Mode of Submission: | on-line via ‘Solent Online Learning’ |
| Number of copies to be submitted: | Where on-line submission via ‘Solent Online Learning’ is used, students are not required to submit a hard copy. |
| Anonymous Marking | No |

# This assignment will NOT be anonymously marked.

**Scenario**

PlacesToStay is a website which allows users to look up local places to stay, such as hotels, hostels and BnBs.. The developers of PlacesToStay would now like to create an accompanying Android app to provide smartphone and tablet users an easy way to look up and display places to stay.

Your task is to develop this app. It has the following requirements:

1. A user should be able to view a map of their current area on their smartphone or tablet. This map should always be positioned at the user's current position on the earth; when the user changes their physical location, the map should reflect this.

2. A user should be able to add a new place to stay at their current location. The user should be able to input the following information:

- name;

- type (hotel, hostel, BnB, etc);

- price per night.

The input should be done using a separate activity. The place to stay should appear on the map as a marker as soon as it has been added.

3. A user should be able to save all places to stay added so far to file, via a menu option.

4. The user should be able to set a preference which determines whether places to stay are saved to file automatically when added. If set, all current places to stay must be saved to file.

5. A user should be able to load all existing places to stay from file. These should be displayed on the map as markers.

6. The user should be able to load places to stay from the web. You should connect to the following web service: (note updated 14/3/20)

https://www.hikar.org/course/ws/get.php?year=20&username=user002&format=XXX

(XXX may be either “json” or “csv”, returning the data as JSON or CSV respectively).

The username=user002 filters out the places to stay so that only places to stay added by the user "user002" are retrieved; you may also supply the value "all" to get all places to stay. **Use the same username you were allocated for the original assignment, or contact me if you did not receive one.**

When the places to stay are loaded they should be shown on the map as markers.

7. If the appropriate preference is set (see requirement 4), a place to stay should be saved to the web after it has been added to the map.

To save to the web, connect to the following web service with a method of POST:

<https://www.hikar.org/course/ws/add.php>(note updated 14/3/20)

The following POST parameters must be specified:

- *username* (your username as described in Requirement 6);

- *name* (the name of the place to stay)

- *type* (the type of the place to stay)

- *price* (the price per night)

- *lat* (the latitude)

- *lon* (the longitude).

- *year*. This must have the value “20”. This is so that the web service knows that it is dealing with data relating to this year’s assignment and not a previous one.

8. Allow the user to see a list of all places to stay in memory so far (loaded in from file, downloaded from the web, or added by the user). When the user selects an entry in this list, the map should be centred on that place to stay. This should be done using fragments, so that either on a tablet or in landscape mode on a phone (your choice), the list appears on the same screen as the map, while otherwise, the list appears in a separate activity.

Ideally, each entry in the list should show not only the name, but also the type (hotel, BnB, hostel, etc), on a separate line in smaller font.

**Tasks**

***I - Coding task***

For each requirement above, **implement** the requirement using the Android API.

***II - Discussion task***

**Discuss** how you have used some or all (as appropriate, depending on the code you have completed) of the following Android topics in your assignment. Include technical detail on how your code using each technology works.

This discussion must relate to your assignment answer rather than being simply a theoretical discussion. It cannot simply be a copy or a paraphrasing of the lecture notes.

The length of discussion required for each topic is variable but should range between 150-400 words.

a) How you have used XML layout files in your application

b) Use of findViewById()

c) Communication between activities using Intents and Bundles, and result codes where appropriate

d) Use of preferences

e) File I/O

f) Network communication, including both GET and POST requests as appropriate

g) JSON or CSV parsing, as appropriate

h) Use of fragments

**Handing in the work**

Please upload a ZIP file containing your full Android Studio project and report. ONLY the code for your answer should be included. Do NOT upload additional source code (e.g. two versions of your project).

**Grading Criteria**

|  |  |  |
| --- | --- | --- |
| **Grade** | **Discussion (30%)**  ***A lack of clarity, or errors in your writeup, will lower your grade appropriately.*** | **Implementation (70%)**  ***Errors in your implementation will lower your grade appropriately.*** |
| F2-F3 | Report does not include discussion topics a) to d) (see the discussion task above), or is unclear or inaccurate. | Requirements 1 to 4 not complete |
| F1 | Discussion of how you have used topics a) to d). There may be a few omissions or errors | Requirements 1 to 4 almost completed; a small number of errors. |
| D | Clear discussion of how you have used topics a) to d). | Requirements 1 to 4 completed. |
| C | Clear discussion of how you have used topics a) to e). | Requirements 1 to 5 completed. |
| B | Clear discussion of how you have used topics a) to f), or topics a) to e) and topic g). For a high B, a clear discussion of how you have used topics a) to g) must be present. | Requirements 1 to 6 completed (1 to 7 for a high B) |
| A | Clear discussion of how you have used topics a) to g), and some discussion on how you have used topic h) (fragments). For a high A, the fragments discussion must be clear and complete. | In addition:  F For an A4, the user should be able to view the list of restaurants as described in Requirement 8. When an entry is clicked, the map should be centred on that restaurant.  For an A3, the list of restaurants should incorporate the restaurant cuisine in a smaller font.  For an A2, in addition to the A4 criteria fragments should be used to produce different layouts (as described in Requirement 8) – there may be a small number of errors with your fragment implementation.  For an A1, all requirements (including Requirement 8) should be complete and working. |

**Other information**

**Copyright**

Please note that using images or text from other websites is infringing copyright and is therefore illegal and not to be done! The only exceptions are if the source website has given you permission, or the material is available in the public domain or under a liberal licence (e.g. Creative Commons). By all means use creativity in the design of your app (though note that you will **not** be given credit for visual design), but use your own material or material you are legally allowed to make use of.

# Learning Outcomes

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the unit descriptors.

# Late Submissions

Students are reminded that:

1. If this assessment is submitted late i.e. within 5 working days of the submission deadline, the mark will be capped at 40% if a pass mark is achieved;
2. If this assessment is submitted later than 5 working days after the submission deadline, the work will be regarded as a non-submission and will be awarded a zero;
3. If this assessment is being submitted as a referred piece of work (second or third attempt) then it must be submitted by the deadline date; any Refer assessment submitted late will be regarded as a non-submission and will be awarded a zero.

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-assessment-principles-and-regulations.pdf?t=1534423842941>

# Extenuating Circumstances

The University’s Extenuating Circumstances procedure is in place if there are genuine circumstances that may prevent a student submitting an assessment. If students are not 'fit to study’, they can either request an extension to the submission deadline of 5 working days or they can request to submit the assessment at the next opportunity (Defer). In both instances students must submit an EC application with relevant evidence. If accepted by the EC Panel there will be no academic penalty for late submission or non-submission dependent on what is requested. Students are reminded that EC covers only short term issues (20 working days) and that if they experience longer term matters that impact on learning then they must contact the Student Hub for advice.

A summary of guidance notes for students is given below:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2p-extenuating-circumstances.pdf?t=1534423896787>

# Academic Misconduct

Any submission must be students’ own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University’s Academic Handbook includes the definitions of all practices that will be deemed to constitute academic misconduct. Students should check this link before submitting their work.

Procedures relating to student academic misconduct are given below:

<http://portal.solent.ac.uk/support/official-documents/information-for-students/complaints-conduct/student-academic-misconduct.aspx>

**Ethics Policy**

The work being carried out by students must be in compliance with the Ethics Policy. Where there is an ethical issue, as specified within the Ethics Policy, then students will need an ethics release or an ethical approval prior to the start of the project.

The Ethics Policy is contained within Section 2S of the Academic Handbook:

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2s-university-ethics-policy.pdf>

**Grade marking**

The University uses a letter grade scale for the marking of assessments. Unless students have been specifically informed otherwise their marked assignment will be awarded a letter grade. More detailed information on grade marking and the grade scale can be found on the portal and in the Student Handbook.

<http://portal.solent.ac.uk/documents/academic-services/academic-handbook/section-2/2o-annex-2-assessment-regulations-grade-marking-scale.pdf?t=1534424273208>

**Guidance for online submission through Solent Online Learning (SOL)**

<http://learn.solent.ac.uk/onlinesubmission>