

ITECH2000 Mobile Development Fundamentals

Assignment 1: ApplInventor App

You will implement in ApplInventor a multi-screen app for a cinema. This app will use a range of components and persist data between executions.

Timelines and Expectations

Percentage Value of Task: 25% (of final semester mark for this course)

Final Due Date: 11:59pm, Sunday 8th September, 2019 (end of Week 8) – after this date, there is a penalty each day.

Cut-off Date: 11:55pm, Sunday 15th September, 2019 – after this date, work will not be accepted

Minimum time expectation: 10 hours

Maximum time expectation: 30 hours – *Most students do not require this much time.*

Learning Outcomes Assessed

This assignment has been designed to enable you to demonstrate your learning in regards to the following outcomes of the course:

- K1. Understand constructs typical of many programming languages such as: variables, expressions, assignment, sequence, selection, iteration, procedures, parameters, return values.
- A1. Design, develop, test and debug mobile apps from a given textual program specification.
- S1. Analyse the input, processing and output needs of small programming problems.
- S2. Design code sequences to realise algorithms in a programming language.
- S3. Design basic user interfaces and develop storyboards to convey designed interaction sequences.

In particular, the assignment has been designed to help give you further experience in using ApplInventor, and to assess your attainment of a range of the learning objectives from weeks 1 through to 6 of the semester.

Getting Assistance and Clarification:

If any part of the task is unclear to you, or you are not quite sure how to do some aspect of the task, you should either contact your lecturer directly (via email, or in person while they are in class), or else post a question to the Discussion Forum on Moodle. However, any questions posted to the forum on Moodle should not include any actual things that you plan to submit (such as screenshots of code you might want to submit), but should instead be asked in a more general way, e.g. asking about a technique, or a similar situation to what you need for the assignment.

Assessment Details

You need to create an app that achieves the described functionality (listed later) for the following scenario:

Scenario:

Your app will be used by the staff who work for a cinema, e.g. HOYTS.

The cinema has a range of movie sessions each day. Customers can book to attend a movie session, and if their circumstances change they can cancel their booking at a later time.

They want you to make an app that can be used to store the booking information and the customer details of the people attending a movie session.

The “Requirements” listed below will help to explain what functionality the app needs to achieve. If you feel you cannot achieve a particular feature, then please try the other features and then come back to it. Remember that you can ask the lecturer questions by email to clarify what is meant by any of the following requirements.

To get full marks you will need to ensure that you have made use of each of the following components or constructs somewhere in your app:

- a Notifier
- a Spinner or ListView (or both)
- a CheckBox
- Labels
- TextBoxes
- Buttons
- A Clock and instants
- HorizontalArrangement or TableArrangement (or both)
- a TinyDB component to gain access to persistent data stored in a database.
- the list construct
- a repetition construct
- a decision construct
- a boolean (AND or OR) expression.
- Procedures you have defined using the ‘to do’ or ‘to do ... result’ blocks.

Please read through all of the requirements *before* you commence the work, so you get a full sense of what is required to be done.

Level 1 – Compulsory Tasks

I expect all students should be able to complete all the requirements in Level 1. By completing these requirements, you should get a 'pass' grade for the assignment, provided you have done things the correct and intended way.

Requirement 1: There must be a welcome screen

This screen should be the first one presented to the user when they start your app – thus it needs to be done on “Screen1”. The screen needs to have mechanisms to get to the various other functions that are offered by the app.

Requirement 2: Obtain information about a customer

It must be possible to enter the details of a new customer into the app. The customer might make several bookings for different movie sessions, or only a single movie session. However, before they can book to attend a movie session, their details need to be input into the app.

The following details need to be input:

- A 5 digit customer ID that does not start with zeros (e.g. 10000 is the smallest acceptable value).
- The name of the customer
- The customer's date of birth
- Whether the customer has a concession card.

There must be a specific screen where the user will enter this information. After validating the input, the screen should pass the inputted information to the prior screen (the screen we were at before we came to this screen for input). This screen should not store the information anywhere – it must return the information to Screen1.

Requirement 3: Let customers book a seat at a movie session

For this assignment, let us assume that each movie theatre has only 5 seats – this will make it easier for you to think about.

To start solving this requirement, assume that there is just one movie: Toy Story at 8pm on Sunday. (See Level 2 requirements for allowing other movies).

When the staff user chooses to book a passenger, the app should show the movie session details (the movie title and time). It should ask which seat to give to the customer. It should provide a way to choose one of the existing customers that have been entered into the app and to record the booking as being for that person.

We need to make sure that the person has not already booked a space at this movie session – they can only book once for each movie session!

For example, pretend that these are the customers already entered in to the system by requirement 2:

- Gary Martin
- Sarah Smith

If Gary has already booked seat #3 in the movie theatre, then he cannot book any of the other 4 seats. If Sarah has not booked any seat yet (but Gary has booked seat #3), then she can book either seat 1, 2, 4 or 5.

When all the details have been provided about who is making the booking, and which seat they want to book, then there must be a way to confirm the booking – to tell the app to store the booking so that we know from now onwards, that the customer has booked that seat in the movie theatre for that session.

We do not want to be entering the customer's personal details here. Requirement 2 is where the customer's personal details are entered. So you must make sure that the customer who wants to make a booking has already been entered into the app before we come here to make the booking.

NOTE: If there are no seats available in the theatre for a movie session, you should warn the user so that they do not try to give a seat to a customer.

Requirement 4: Allow customers to cancel a booking

A customer is allowed to cancel their booking. You need to provide a way in the app for removing the customer's booking – the seat in the movie theatre will become available to any customer (including the same customer if they choose to book a seat in the future).

Requirement 5: Show summary information about a movie session

There must be a screen which will display summary information about a movie session. It needs to tell the user the following things:

- The name of the movie being played.
- The time that the movie will be played.
- The number of people booked to see the movie.
- The number of seats that are available to book.
- The number of people with a concession card who are attending this session.
- The number of children who are attending this session (e.g. customers who are under 18 years old).

Requirement 6: Information about customers details, must persist between separate uses of the app

The details about customers from requirement 2, should only ever need to be entered once. If I enter Gary Martin's details today, close the app, restart the phone tomorrow, and start the app Gary Martin's details should still be known to the app – we should still be able to book a seat on a flight for Gary Martin.

You should use TinyDB to achieve this. Do not use a File.

Requirement 7: Information about customers and their bookings must persist between separate uses of the app

The details of all the bookings, made during requirement 3, and possibly changed by requirement 4, need to persist. For example, if today I book Gary Martin and Sarah Smith to attend the movie, close the app, restart my phone tomorrow and restart the app, they should both still have bookings for the same seats as today. If Sarah Smith cancels her seat tomorrow, then when I start the app the day after tomorrow, it should still show that Gary Martin has his seat booked, but Sarah Smith's seat should be available.

You should use a File for this. Do not use a TinyDB.

Level 2 – More complicated Tasks

Some students may find these tasks harder. We do not expect everyone to achieve all of these tasks, but if you do, you will probably receive a higher grade. Before you attempt any Level 2 tasks, you should finish all Level 1 tasks.

Level 2 is about allowing the app to work for multiple, different movie sessions. For example, one movie session could be for Aladdin at 9am, another movie session could be Men in Black: International at midday, and another could be Aladdin at 1.30pm.

Requirement 8: Create a new movie session

To achieve this requirement you must let the user create a new movie session. They must specify the title, day and time of the movie session. They must be able to specify the name of the file which will be used to store the bookings details for that movie session. The customer details (such as their date of birth or concession status) should still be stored in the TinyDB.

Requirement 9: Choose the movie session to be the “current” one to work with.

Because there are now multiple movie sessions, you need to let the user switch between one movie session and another. There must be a way that the user can change which movie session is the one which is currently being changed / viewed. You can assume that the user will be required to provide the name of the file that contains the bookings data.

Requirement 10: Ensure that there is a ‘default’ “current” movie session for the app

The first time the app is run (e.g. by the person who grades you), they will need to have the app thinking it is dealing with a movie session already, such as was the case for level 1. Therefore, you need a way to ensure that there will be a “current” movie session when the app is run the first time – if the user has to do something to ensure this, make sure it is obvious to the user.

Requirement 11: You must have full validation of all input

It should not be possible anywhere in the app, for the user to enter data that does not make sense, or which is incorrect, or which is inconsistent with the described requirements. This includes for input argument slots of procedures. Your marker will test the robustness of your app so it shouldn't crash if it receives “unexpected” input.

Level 3 – Advanced Tasks

These tasks are hard to achieve, and should only be attempted if you completed Level 2. You need to achieve these tasks if you want the HD (High Distinction) grade because they require you to apply problem solving skills without any guidance.

Advanced-Requirement A: User should not need to enter file names

After creating a new movie session (see requirement 8), the user should never again need to type in the file name, nor even see or remember it. Instead, they should only need to know the movie title and the time it is showing.

Advanced-Requirement B: User should be able to choose the movie session on the screen for making a booking of a seat or cancelling a booking.

On the screen used for requirement 3, and the screen used for requirement 4 (they may be the same or different screens depending on how you have done it), the user should be able to select the movie session there, instead of needing to separately choose a “current” movie session (as in Requirement 9).

Advanced-Requirement C: Report on a particular customer’s current bookings

There should be a screen where you can choose a particular customer and it will tell us all the movie sessions that they have booked a seat for. If they cancelled a booking, it should not be shown on this screen.

Note: You may include images in your app if you feel that will help, but there are no marks given for their use/inclusion.

Plagiarism:

Plagiarism is the presentation of the expressed thought or work of another person as though it is one's own without properly acknowledging that person. You must not allow other students to copy your work and must take care to safeguard against this happening. More information about the plagiarism policy and procedure for the university can be found online at:

<http://federation.edu.au/students/learning-and-study/online-help-with/plagiarism>.

Submission

You must “export” your AppInventor project in order to submit it through Moodle. To do this, go to “Projects” menu, and select “Export selected project (.aia) to my computer”.

Upload the file to Moodle through the assignment link labelled “Submit Assignment 1”. This link will only become available after you have completed the ‘Declaration of Originality’ form for the assignment, which requires you to accept the Student’s Statement – which is instead of you needing to sign a paper cover sheet. It is a legal declaration that the work was done by you, without any part of it being done by someone else.

Feedback

You can expect to receive your final mark and feedback comments within 2 weeks of the due date, or the date which you submitted your work, whichever is later.

You may be required to attend an interview with the marker to answer questions about your work before they will mark your submission.

Marking Criteria

You will be assessed according to the following scheme:

Requirement 1 – There must be a welcome screen Requirement 1 has been completed – it is included in the submitted app Requirement 1 has been achieved in the most appropriate way.	1 mark 1 mark
Requirement 2 – Obtain information about customer Requirement 2 has been completed, and behaves correctly Requirement 2 has been achieved in the most appropriate way	1 mark 1 mark
Requirement 3 – Let customers book a seat at a movie session Requirement 3 has been completed and behaves correctly for correct uses Requirement 3 behaves correctly for incorrect uses Requirement 3 has been achieved in the most appropriate way	1 mark 1 mark 2 marks
Requirement 4 – Allow customers to cancel a booking Requirement 4 has been completed and behaves correctly Requirement 4 has been achieved in the most appropriate way	1 mark 2 marks
Requirement 5 – Show summary information about a movie session Requirement 5 has been completed and behaves correctly Requirement 5 has been achieved in the most appropriate way	1 mark 2 mark

<p>Requirement 6 – Persist customer information using TinyDB</p> <p>Requirement 6 has been completed and behaves correctly</p> <p>Requirement 6 has been achieved in the most appropriate way</p>	<p>1 mark</p> <p>2 mark</p>
<p>Requirement 7 – Persist booking information using a File</p> <p>Requirement 7 has been completed and behaves correctly</p> <p>Requirement 7 has been achieved in the most appropriate way</p>	<p>1 mark</p> <p>2 mark</p>
<p>Requirement 8 – Create a new movie session</p> <p>Requirement 8 has been completed and the app behaves correctly with its inclusion</p> <p>Requirement 8 has been achieved in the most appropriate way</p>	<p>0.5 marks</p> <p>0.5 marks</p>
<p>Requirement 9 – Choose “current” movie session to work with</p> <p>Requirement 9 has been completed and the app behaves correctly with its inclusion</p> <p>Requirement 9 has been achieved in the most appropriate way</p>	<p>1 mark</p> <p>1 mark</p>
<p>Requirement 10 – Ensure that there is a “default” movie session</p> <p>Requirement 10 has been completed and the app behaves correctly with its inclusion</p> <p>Requirement 10 has been achieved in the most appropriate way</p>	<p>1 mark</p> <p>1 mark</p>
<p>Advanced-Requirement A – User should not need to enter file names</p> <p>Advanced-Requirement A has been completed correctly and the app behaves correctly with its inclusion</p> <p>Advanced-Requirement A has been achieved in the most appropriate way</p>	<p>1 mark</p> <p>2 marks</p>
<p>Advanced-Requirement B – User can choose movie session on booking/cancel screen</p> <p>Advanced-Requirement B has been completed correctly and the app behaves correctly with its inclusion</p> <p>Advanced-Requirement B has been achieved in the most appropriate way</p>	<p>1 mark</p> <p>3 marks</p>
<p>Advanced-Requirement C – Report on particular customer’s bookings</p> <p>Advanced-Requirement C has been completed correctly</p> <p>Advanced-Requirement A has been achieved in the most appropriate way</p>	<p>1 mark</p> <p>2 marks</p>
<p>Appropriate Components</p> <p>Has selected appropriate components for input and for output of information across all screens.</p> <p>Has used TextBoxes, CheckBoxes, Labels, and Buttons appropriately (including using their action blocks and event blocks) appropriately</p> <p>Has used a Spinner or ListView (including their action blocks and event blocks) appropriately</p> <p>Has used the screen-changing mechanisms correctly and appropriately</p> <p>Any other visual components used have not been used in an inappropriate manner</p>	<p>1.5 marks</p> <p>0.5 marks</p> <p>0.5 marks</p> <p>0.5 marks</p> <p>1 marks</p> <p>0.5 marks</p>
<p>Programming Constructs</p> <p>Has used repetition construct to repeat code sequences appropriately and in appropriate places</p> <p>Has used decision constructs appropriately / in appropriate places</p>	<p>2 marks</p>

Has formed appropriate boolean and relational expressions	2 marks
Has used the list construct for appropriate purposes and used its action blocks in an appropriate manner to manipulate/use lists.	2 marks
Has formed procedures appropriately and in appropriate circumstances	2 marks
Has used the TinyDB component appropriately	2 marks
Has used the Clock component and “time instant” values appropriately	2 marks
	1 mark
Names of variables, components, procedures, etc.	
The names used for variables, components, procedures and their argument slots, are the appropriate, and are chosen to clearly communicate their purpose in the code.	2 marks
Requirement 11	
Validation is implemented correctly and thoroughly throughout the app.	3 marks
Overall ability to construct code sequences, in response to appropriate events, to achieve problem solving of computer programming tasks	3 marks
TOTAL:	60 marks

The final mark out of 60, will be scaled to a value out of 25 to contribute an amount towards your semester total marks.