FACULTY OF SCIENCE AND

ENGINEERING SEMESTER 2, 2018

IAB330: Mobile App Development

Assignment 3: App Prototype

Due Date: Friday, 2nd Nov 2018, 11:59 pm

Assignment submission as a team through Blackboard

Weight: 50%

You must sign below. By signing this form, you agree to the following:

We declare that all of the work submitted for this assignment is our own original work except for material that is explicitly referenced and for which we have permission, or which is freely available (and also referenced)

The assignment shall be conducted in a team of 3-4 students, each team member must sign, as it is a formal agreement that represents that everyone is contributing to the whole assignment.

<table>
<tr><th colspan="3">Team Member Details</th></tr>
<tr><th>Student Number</th><th>Student Name</th><th>Signature</th></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

Task 1: Final Prototype (30 marks)

Submit the final version of your mobile application (developed using Xamarin).

Submit the source code via a Git repository link that is publicly accessible.

Your source code should compile on the lab environment (Android project, Visual Studio 2017 on Windows) and run in the **Android** emulator. If your app needs additional packages, please provide a list of the required packages and their versions. We recommend to use cloud-based services for any remote databases, as opposed to local ones for better evaluation of your code and app functionality by the teaching team.

Task 2: Report (10 marks)

Submit a report that includes elaborations on the following aspects:

User Stories

Provide a revised list of your MVP user stories from Assignment 2 and indicate their implementation status.

Include any nice-to-have features you may have implemented.

If you did not manage to complete the MVP implementation, explain how you intended to implement the missing features.

User Interface

Include screenshots of the primary screens of your app, and explain their functionality.

Software Architecture

Provide a diagram (e.g. UML) of your entire software architecture. Discuss how your architecture changed/evolved compared to the one proposed in Assignment 1 during the implementation process. Explain the reasons behind the changes and what factors influenced the evolution of your software achitecture.

Testing and Quality Assurance Strategy

Explain which testing methods you used and how you applied them in order to assure the quality of your app.

Reflection on Learning

Summarize the faced challenges and difficulties and how your team resolved them.
Focus on technical, project management, design and/or teamwork challenges.

Elaborate on what you learned during this semester as a team.

Task 3: Presentation Demo (10 marks)

You will demonstrate a demo of your project in Week 13 to the audience and teaching team using the code implemented so far. In the presentation, you are required to explain the purpose of your app clearly and concisely, and showcase what you achieved by demonstrating the features you have implemented in your app using the emulator. At the end of your presentation, you will be asked a few technical questions related to your project. The duration of your presentation should be of **maximum 8 minutes**.

Presentation day: Friday, 26th of October, at GP-C405

IAB330 Assessment Criteria: Assignment 3 - Mobile App Prototype

CRITERION	MA RKS	7 85%-100%	6 75%-84%	5 65%-74%	4 50%-64%	Refer / Fail 0%-49%
Task 1: Final Prototype Submit the final version of your mobile application (developed using Xamarin). Your source code should compile on the lab-programming environment (Android project, Visual Studio 2017 on Windows) and run in the **Android** emulator. If your app needs to install additional packages, please include a list of the required additional resources and their versions. **Submit the source code** via a Git repository link that is publicly accessible.	/30	Defect-free implementation of *all MVP features* of the project. The submitted source code compiles and runs in an Android emulator. All required resources are listed. (20 marks) Source code is well structured in a MVVM (**preferred**) or MVC structure and makes excellent and appropriate use of software design patterns. (5 marks) Source code is very well formatted, very easy to read and understand, and makes efficient use of computational resources (e.g. no 'spaghetti code', appropriate separation of code into distinct methods minimising duplication, complex procedures have explanatory comments, avoids long statement chains, minimises use of global variables, uses appropriate binding options, …). (5 marks)	Very good, almost defect-free implementation of *all MVP features* of the project. The submitted source code compiles and runs in an Android emulator. All required resources are listed. Source code is well structured in a MVVM (**preferred**) or MVC structure and makes very good and appropriate use of software design patterns. Source code is well formatted, easy to read, understand and efficient.	Good, mostly defect-free implementation of *most MVP features* of the project. The submitted source code compiles and runs in an Android emulator. All required resources are listed. Source code is well structured in a MVVM (**preferred**) or MVC structure and makes good and sensible use of software design patterns. Source code is mostly well formatted, and quite easy to read and understand.	Implementation of some *basic MVP features* of the project. The submitted source code compiles and runs in an Android emulator. Source code is well structured in a MVVM (**preferred**) or MVC structure and makes appropriate use of some software design patterns. Source code is not well formatted and difficult to read and understand.	Failed to implement *basic MVP features* of the project. The submitted source code does not compile and does not run in an Android emulator. Failed to submit the source code.

Task 2: Report **User Stories** Provide a revised list of your MVP user stories and indicate their implementation status. Include any implemented nice-to-have features. If you did not manage to complete the MVP implementation, explain how you intended to implement the missing features.	/10	**User Stories** List with all MVP features is provided, and the status of each is indicated. All additionally implemented features are included. The implementation approach for missing ones is very well explained. (1 mark)	**User Stories** List with all MVP features is provided, and the status of each is indicated. All additionally implemented features are included. The implementation approach for missing ones is well explained.	**User Stories** List with all MVP features is provided, and the status of each is indicated. All additionally implemented features are included. The implementation approach for missing ones is somewhat explained.	**User Stories** List with all MVP features is provided, and the status of each is indicated. All additionally implemented features are included. Implementation approach for missing features not provided.	**User Stories** Missing or inappropriate list of MVP user stories. No indication of completion status. No discussion of missing features and their implementation strategy.
User Interface Include screenshots of the primary screens of your app, and explain their functionality.		**User Interface** Screenshots of each major page with excellent explanation of the functionality. (1 mark)	**User Interface** Screenshots of each major page with very good explanation of the functionality.	**User Interface** Screenshots of most major pages with good explanation of the functionality	**User Interface** Screenshots of most major pages with some explanation of the functionality.	**User Interface** Missing screenshots of major pages or missing explanation of the functionality
Software Architecture Provide a diagram (e.g. UML) of your entire software architecture. Compare how your architecture changed/evolved compared to the one proposed in Assignment 1 during the implementation process. Discuss the reasons behind the changes and what factors influenced the evolution of your software achitecture.		**Software Architecture** Excellent diagram that includes all the classes. (1 mark) Excellent comparison between the proposed and final software architecture. (2 marks) Exceptional discussion of the reasons and factors that led to the changes and evolution of the software architecture. (2 marks)	**Software Architecture** Very good diagram that includes all the classes. Very good comparison between the proposed and final software architecture. Very good discussion of reasons and factors for the change and evolution of the software architecture.	**Software Architecture** Good diagram that includes most of the classes. Good comparison between the proposed and final software architecture. Good discussion of reasons and factors for the change and evolution of the software architecture.	**Software Architecture** Acceptable diagram that includes some of the classes. Acceptable comparison between the proposed and final software architecture. Acceptable discussion of reasons and factors for the change and evolution of the software architecture.	**Software Architecture** Insufficient or missing diagram, comparison and discussion of the software architecture.
Testing and Quality Assurance Strategy Explain which testing methods you used and how you applied them in order to assure the quality of your app		**Testing and Quality Assurance Strategy** Excellent explanation of the applied testing and quality assurance strategy (1 mark)	**Testing and Quality Assurance Strategy** Very good explanation of the applied testing and quality assurance strategy	**Testing and Quality Assurance Strategy** Good explanation of the applied testing and quality assurance strategy	**Testing and Quality Assurance Strategy** Sufficient explanation of the applied testing and quality assurance strategy	**Testing and Quality Assurance Strategy** Insufficient or missing explanation of the applied testing and quality assurance strategy
Reflection on Learning Summarize the faced challenges and difficulties (technical, project management, design, team work) and how your team resolved them. Elaborate on what you learned during this semester as a team		**Reflection on Learning** Excellent elaboration on the learning and summary of challenges and their solutions. (2 marks)	**Reflection on Learning** Very good elaboration on the learning and summary of challenges and their solutions	**Reflection on Learning** Good elaboration on the learning and summary of challenges and their solutions	**Reflection on Learning** Sufficient elaboration on the learning and summary of some challenges and their solutions	**Reflection on Learning** Missing elaboration on the learning and summary of some challenges and their solutions

Task 3: Project Demo **Aspects of presentation:** - Presentation of app purpose (elevator pitch) - Presentation of achieved work - Q&A - Quality of the presentation	/10	Presentation clearly and succinctly articulates the purpose of the app. (2 marks) Presented demo shows the functionality of 90-100% of the *MVP* features. (5 marks) Answering clearly to all the questions. (1 marks) Presentation is coherent and compelling. (2 mark)	Presentation articulates the purpose of the app well. Demonstrated demo shows the functionality of 70-80% of the *MVP* features. Provided clear answers to most of the questions. Presentation is coherent and covers all the relevant content.	Presentation articulates purpose of the app. Demonstrated demo shows the functionality of most (more than 50%) of the *MVP* features. Provided clear answer to some of the questions. Presentation covers most of the relevant content.	Purpose of the app somewhat articulated. Demonstrated demo shows the functionality of 50% of the *MVP* features. Unable to answer most of the questions. Presentation covers some of the relevant content.	Purpose of the app not articulated. Missing presentation demo, or the demo does not show basic *MVP* features. Failed to answer any questions. Presentation is missing most of the relevant content.
TOTAL	/50					