IAB330 Assessment Criteria: Assignment 2 - Mobile App Prototype

(Assessment Item 4)

CRITERION	MARKS	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). The app should implement one of the two Project options that you designed in Assignment 1 (Mobile App Design). Your source code should compile on the lab programming environment (Android project, Visual Studio 2017 on Windows) and run in the **Android** simulator. If your app needs to install additional packages, please include a list of the required additional resources. **Submit the source code** in a compressed (zip) file that contains the entire project. **Record a presentation demo** of your app and include an easily accessible URL link. The link should not require an account or login.	/30	Defect-free implementation of *all required features* of the project. The submitted source code compiles and runs in an Android simulator. All required resources are listed. (15 marks) Recorded presentation demo shows the functionality of all *required* features. (5 marks) Implemented and recorded presentation demo of the functionality of at least one *'nice-to-have'* feature. (4 marks) Source code is well structured in a Model/View/ViewModel stucture. (2 marks) Source code is very well documented. (2 marks) Source code is well formatted, easy to read and easy to understand (procedures not too long, not exceeding number of parameters, avoid long statement chains, minimize use of global variables). (2 marks)	Very good, almost defect-free implementation of *all required features* of the project. The submitted source code compiles and runs in an Android simulator. All required resources are listed. Recorded presentation demo shows the functionality of 70-80% of the *required* features. Source code is well structured in a Model/View/ViewModel stucture. Source code is well documented. Source code is well formatted, easy to read and easy to understand (procedures not too long, not exceeding number of parameters, avoid long statement chains, minimize use of global variables).	Good, mostly defect-free implementation of *most required features* of the project. The submitted source code compiles and runs in an Android simulator. All required resources are listed. Recorded presentation demo shows the functionality of most (more than 50%) of the *required* features. Source code is well structured in a Model/View/ViewModel stucture. Source code is documented. Source code is well formatted.	Implementation of some *basic required features* of the project. The submitted source code compiles and runs in an Android simulator. Recorded presentation demo shows the functionality of 50% of the *required* features. Source code is well structured in a Model/View/ViewModel stucture. Source code is not sufficiently documented, is not well formatted and difficult to read.	Failed to implement *basic required features* of the project. The submitted source code does not compile and does not run in an Android simulator. Missing recorded presentation demo, or the demo does not show basic *required* features. Failed to submit the source code.

Task 2: Report **User Interface** Include screenshots of the user interface of each page, and explain the functionality of your app. Explain how your app implements each of the required and 'nice-to-have' features listed in the project specification. If you did not manage to complete the implementation, explain how you intended to implement the missing features. Elaborate on the applied UI patterns, and explain your design considerations. **Software Architecture** Explain the software architecture of your implementation Elaborate on the applied software architectural patterns Explain the implemented functionality of each component and class **Testing and Quality Assurance Strategy** Explain how you assured the quality of your application Explain your testing methodology **Reflection on Learning** Elaborate on what you learned during this semester as a team Summarize the faced challenges and difficulties and how your team solved them	/10	**User Interface** Screenshots of each page with good explanation of the functionality (1 mark) Exceptional explanation how the app implements each feature, or how the implementation was intended (if not finished) (2 marks) Good elaboration on the applied UI patterns and design considerations (1 mark) **Software Architecture** Good explanation of the software architecture (1 mark) Applied software architectural patterns are well explained (1 mark) Good explanation of the implemented functionality of each component and class (2 marks) **Testing and Quality Assurance Strategy** Good explanation of the applied testing and quality assurance strategy (1 marks) **Reflection on Learning** Good elaboration on the learning and summary of challenges and their solutions (1 marks)	**User Interface** Screenshots of each page with good explanation of the functionality Very good explanation how the app implements each feature, or how the implementation was intended (if not finished) Some elaboration on the applied UI patterns and design considerations **Software Architecture** Good explanation of the software architecture Good explanation of the implemented functionality of each component and class **Testing and Quality Assurance Strategy** Good explanation of the applied testing and quality assurance strategy **Reflection on Learning** Good elaboration on the learning and summary of challenges and their solutions	**User Interface** Screenshots of each page with good explanation of the functionality Good explanation how the app implements each feature, or how the implementation was intended (if not finished) **Software Architecture** Good explanation of the software architecture **Testing and Quality Assurance Strategy** Explanation of the applied testing and quality assurance strategy **Reflection on Learning** Good elaboration on the learning and summary of challenges and their solutions	**User Interface** Screenshots of most pages with some explanation of the functionality Explanation how the app implements each feature, or how the implementation was intended (if not finished) **Software Architecture** Sufficient explanation of the software architecture **Testing and Quality Assurance Strategy** Sufficient explanation of the applied testing and quality assurance strategy **Reflection on Learning** Sufficient elaboration on the learning and summary of some challenges and their solutions	**User Interface** Missing screenshots of pages or missing explanation of the functionality Missing explanation on how the app implements each feature, or how the implementation was intended (if not finished) **Software Architecture** Insufficient or missing explanation of the software architecture **Testing and Quality Assurance Strategy** Insufficient or missing explanation of the applied testing and quality assurance strategy **Reflection on Learning** Missing elaboration on the learning and summary of some challenges and their solutions
TOTAL	/40					