Criterion	7	6	5	4	3	2-1
Deployment and basic operation, layout and usage of the Application. **(10 Marks)**	The application successfully implements the agreed functionality using the nominated APIs and services and there has been a successful deployment to the public cloud (AWS, Azure or GCP). API usage is authenticated as necessary. The application appears robust and handles error conditions gracefully. The basic UI is clean, uncluttered and functional.	The application successfully implements most of the agreed functionality and has been successfully deployed to the public cloud. The application is generally robust and handles error conditions gracefully. The UI is mainly clean and uncluttered and functional.	The application successfully implements most of the agreed functionality, but there may be minor limitations or errors or a lack of robustness. The UI is mainly clean but there may be errors of alignment or styling.	The application successfully implements the majority of the agreed functionality, but there may be significant limitations. The application may exhibit errors and a lack of robustness. The UI is mostly clean, but there may be some clutter and weak design	The application may have been deployed to the cloud, but the functionality and usage of APIs and services is unsatisfactory or limited. The application is not robust and error handling is weak. Basic UI is cluttered and poorly designed.	The application does not meet the specification or is fundamentally flawed in its use of the data sources and services. Robustness is non-existent and errors are frequent.
A professional demo and report (6 + 4 = 10 marks) **[Note that the marks for the scaling aspects of the demo and description are covered later. This is the general standard of the demo and report]**	The report is thoroughly professional and addresses each of the listed requirements in detail and with only occasional errors of grammar or structure. The technical discussion is of a high standard, concisely describing the architecture and implementation of the system and the experiences during development. Appropriate use is made of diagrams and small code fragments to enhance understanding. The test plan is consistent with the requirements and exhibits a suitable number of test cases for the application. The demo is well structured, and successfully presents use cases consistent with the report. The presenter is thoroughly familiar with the application and its use.	The report is reasonably professional and addresses most listed requirements. Errors of grammar or structure may be more common. The technical issues are covered in the report but the discussion lacks some precision or depth. Diagrams and perhaps code fragments are present but they could be better chosen. The test plan is consistent with the requirements and exhibits a suitable number of test cases for the application. The demo is well structured, and successfully presents use cases consistent with the report. The presenter is reasonably familiar with the application and its use.	The report addresses most listed requirements, but is not as well organized. Errors of grammar or structure are more common. Technical issues are covered but the approach often lacks depth, Diagrams and code fragment may be poorly chosen. The test plan is consistent with the requirements but the coverage may be limited. The demo is reasonably well structured, and presents use cases consistent with the report, which are mostly successful. The presenter successfully uses the application but the demo isn't especially slick.	The report is adequate, but the coverage is deficient in a number of the listed requirements and grammar and structure are variable. The technical discussion is weak and superficial. Important aspects are missed or covered poorly and the diagrams and code fragments may not be present at all. The test plan is present but inadequate. There are insufficient test cases. The demo is somewhat disorganized, but still manages to cover the main use cases. The presenter successfully uses the application but appears disorganized.	The report is somewhat adequate, but the coverage is deficient in many of the listed requirements. Grammar and structure are of variable quality. Technical discussion is inadequate or missing. There is no architecture diagram or use of code fragments. The test plan is poor or not present at all. The demo is poorly prepared and organized. There is some coverage of the application use cases, but the presenter is not competent in the use of the application.	The report is flawed and doesn't meet the requirements. There may be whole sections missing or poorly covered and the structure is not coherent. The demo is poorly structured and the presenter disorganized. The use cases are not presented adequately.
Marks (20)						

Criterion	7	6	5	4	3	2-1
Architecture and Persistence **(10 Marks)**	The application architecture is stateless and demonstrably supports scaling of the application In response to load variations. Persistence is chosen to align closely with the application and its architecture, and acts to support application usability and to limit latency. There are at least two levels of persistence service supported.	The architecture is stateless and plausibly supports elastic scaling. Persistence aligns with the application and its architecture, and generally supports application usability and limits latency. At least two levels of persistence are supported.	The application architecture is stateless and offers some support for elastic scaling.. Persistence is reasonably well-aligned, but some choices may introduce unexpected latency or scale poorly to larger data sets. At least two levels of persistence are supported.	The architecture is ad hoc, and not necessarily stateless. Support for scaling is unclear. Persistence is included, but the services are poorly chosen and are incomplete or not well aligned with the application or its architecture. At least one persistence service is supported.	There is no obvious architectural support for scaling and the approach is not stateless. Persistence is not successfully implemented or inappropriate. .	The application architecture is deeply flawed and there is no evidence of persistence.
Code Quality and Approach to Development **(5 Marks)**	The application shows clear evidence of a professional approach to development and code quality. For example: commenting is sparse and non-redundant; whitespace is used consistently and in accord with the examples. Indentation is consistent and appropriate.	There is good evidence of a professional approach. There may be some modest inconsistencies in whitespace or alignment or commenting.	There is some evidence of a professional approach, but code quality may be variable, with more significant violations.	There is limited evidence of a professional approach and code quality is weak with frequent and serious violations	Application development has been ad hoc and code quality is very weak.	Application development is deeply flawed and code quality is poor.
Application Scaling and Demonstration. **(15 marks)**	The application successfully demonstrates automated API or policy based scaling in response to varying load consistent with the architectural choice. The scaling metrics and KPIs are an extremely good fit for the application and architecture. Resources are allocated and deallocated smoothly and appropriately. Latencies are modest or not readily observable. Except for the transition periods, the user remains unaware of the provisioning.	The application successfully demonstrates automated elastic scaling. Metrics and KPIs are a good fit for the application and its architecture. Resources are allocated and deallocated fairly smoothly and appropriately. Latencies are at times excessive and the response to variations in load is not optimal. Users are generally not aware of the issues.	The application successfully demonstrates automated elastic scaling. Scaling metrics and KPIs are reasonably well chosen, but their operation is not well tuned. Resource allocation and deallocation may not be smooth. Latencies may be excessive and users are often aware of the issues.	The application does not support automated elastic scaling. The submission however includes evidence of manual provisioning in response to variations in load and subsequent successful use behind a load balancer. Latencies are frequently excessive and Users are well aware of the issues.	The submission does not demonstrate automated scaling or provide instructions to manually provision resources. Some effort has been made to support load balancing with a fixed pool.	The submission does not address the issue of scaling in any coherent or adequate fashion.
Marks (30)						
TOTAL (50)						

CAB432 Assignment 2: Group Task CRA Rubric. Overall weighting 50% of the semester marks.