# SOFTENG 325 Assignment 1 Report

The task assigned to us was to build a booking service application for a small venue with a capacity of three-hundred and seventy-four seats. According to the system brief, the service should be able to handle information about concerts to be held at the venue and the various performers who would perform at the venue. The service should also store user information and allow them to book seats for concerts.

The service is expected to experience high load when concert tickets go on sale and as such, scalability was a major issue that had to be accounted for when building the application. Several methods were used to improve the scalability of the system including the use of REST, use of cookies, the use of JPA, image caching and good domain model design.

The service is written by following the REST protocol meaning that it is entirely stateless. This allows for us to use replica servers because there is no information stored on the actual server regarding the user. Because we are able to use replica servers, the server is much more scalable.

The use of cookies enables us to identify which client is connecting to the service and allows us to store specific information regarding this client in a database. This is what allows us to have a stateless service which enables us to use replica servers.

JPA was used to build the service. In order to increase scalability, the Hibernate implementation of JPA was used. Hibernate avoids loading objects from the database until the object is needed for use by the application thus reducing the impact on memory. By doing this, the risk of running out of memory when the server is under high-load has been reduced thereby increasing scalability.

The client is able to get images of the performers. In order to reduce the load on the server, whenever a client tries to get a performer image, the image is cached on the client-end. This means that the client does not need to send a new GET request to the server every time they want an image. Because this reduces server load, scalability is increased.

The domain model was designed in a way that allows for efficient querying. It was ensured that every object in the database has a unique identification value meaning that it is very fast and very easy to find any given object in the database. The identification value is stored along with all the other information pertaining to an object as a row in the entity table. Links between objects are represented using separate tables which link the identification values of each object which allows us to quickly rebuild entire objects from the database. This technique also allows us to save memory as we are not creating an entirely new table with all the information pertaining to two different objects. Due to these reasons, the domain model allows for increased scalability.

In conclusion, the system fulfils the requirement of being scalable to an extent. However, there are other methods that could be used to further improve scalability that have not been implemented. For example, caching could be implemented on GET requests that return large amounts of information, such as getPerformers and getConcerts. We could also implement database locking in order to avoid concurrency issues. These methods should be implemented in the future in order to ensure that the system is able to withstand the load required in the brief.