What is Tokeneer?

- Software system for controlling access to a high-security area
- Developed so that security and other requirements could be formally verified
- Requirements specification is contained on a large document
  - Approach is mainly through Use Cases

Tokeneer Requirements Specification

- **Functional (behavioral) requirements through eleven (11) Use Cases**
  - Each use case covers a key scenario
- **Some supplementary diagrams as well**
  - Some for non-functional requirements

5 Behavioural Requirements

The required behaviour of the ID Station is specified here using cases, which run through formalised use cases of the ID Station and define the interaction between the ID Station and the controlled system. In other words, a scenario describes a number of actions the user must perform to achieve a goal. The technical details of exactly how these actions are performed are not discussed, but it is implied that all actions have been tested such that the system will not allow the user to access a high-security area if access is not granted.

**Environmental and Robustness Requirements**

Here is a timing Requirement. Should it be Said better?

**Functional Requirements** (when combined with assumptions)
Robustness requirements

Anticipating Requirements Validation and Design

There is still a need to coordinate and integrate what these scenarios say
Use of Diagrams to Augment the Use Cases

- Not all of our requirements types were addressed by these use cases
- Some of them are addressed separately from the use cases
- Some diagrams are redundant with use cases (to support cross checking)
- Some diagrams could/should be embedded in some of the use cases

Artifacts are needed in order to specify functional/behavioral requirements
Augments Functional Rqmt.

Security Requirement?

6 Design Constraints
The system will be developed and run on an x86-based operating system.