|  |
| --- |
| Software Requirements Specification |
| GLOBLOCK |
| BSHCE BSc in Computing |

|  |
| --- |
| Alex Quigley - x10205691  10/27/2013 |

# Document Control

## Project Information

|  |  |
| --- | --- |
| **Project Title:** | Globlock |
| **Course Code:** | BSHCE4 |
| **Course Title:** | BSc (Hons) in Computing |
| **Course Specialisation:** | Network and Mobile Technologies |
| **Student Name:** | Alex Quigley |
| **Student Number:** | 10205691 |
| **Date:** | 06th October 2013 |

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Scope** | **Prepared** | **Date** |
| 1.00 | Create | Alex Quigley | 21/10/2013 |
| 1.01 | Update to Structure/Headings | Alex Quigley | 22/10/2013 |
| 1.02 | Added Use Cases | Alex Quigley | 24/10/2013 |
| 1.03 | Update Use Cases and Diagrams | Alex Quigley | 25/10/2013 |
| 1.04 | Added Non Functional Requirements | Alex Quigley | 26/10/2013 |
| 1.05 | GUI and Final Proofing | Alex Quigley | 27/10/2013 |

## Distribution List

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Title** | **Version** | **Date** |
| Paul Hayes | Project Supervisor | 1.05 | 27/10/2013 |
| Jonathan McCarthy | Module Co-ordinator | 1.05 | 27/10/2013 |

Table of Contents

[Document Control 1](#_Toc370600412)

[Project Information 1](#_Toc370600413)

[Revision History 1](#_Toc370600414)

[Distribution List 1](#_Toc370600415)

[Introduction 5](#_Toc370600416)

[Purpose 5](#_Toc370600417)

[Intended Audience 5](#_Toc370600418)

[Project Scope 5](#_Toc370600419)

[Overall Description 7](#_Toc370600420)

[Product Perspective 7](#_Toc370600421)

[Product Features 7](#_Toc370600422)

[User Account Management and Permissions 7](#_Toc370600423)

[Repository and Document Management 8](#_Toc370600424)

[Globe Management 8](#_Toc370600425)

[Reporting 8](#_Toc370600426)

[User Classes, Characteristics and Perspective 8](#_Toc370600427)

[Super User 8](#_Toc370600428)

[Regular user 8](#_Toc370600429)

[Operating Environment 9](#_Toc370600430)

[Globes 9](#_Toc370600431)

[Block Device 9](#_Toc370600432)

[Windows Service and Operating System 9](#_Toc370600433)

[Management Client Software 9](#_Toc370600434)

[Web Server & Database 9](#_Toc370600435)

[Application Programming Interface 9](#_Toc370600436)

[NFC Device 9](#_Toc370600437)

[Design and Implementation Constraints 10](#_Toc370600438)

[User Documentation 10](#_Toc370600439)

[Definitions, Acronyms and Abbreviations 10](#_Toc370600440)

[System Features 11](#_Toc370600441)

[System Feature 1.0: Account Management 11](#_Toc370600442)

[Functional Requirement 1.1: Create Account 11](#_Toc370600443)

[Functional Requirement 1.2: Login 11](#_Toc370600444)

[System Feature 2.0: User Management 12](#_Toc370600445)

[Functional Requirement 2.1 Create Group 12](#_Toc370600446)

[Functional Requirement 2.2: Create User 12](#_Toc370600447)

[Functional Requirement 2.3: Modify Group 13](#_Toc370600448)

[Functional Requirement 2.4: Modify User 13](#_Toc370600449)

[System Feature 3.0 Document Management 14](#_Toc370600450)

[Functional Requirement 3.1: Upload Document 14](#_Toc370600451)

[Functional Requirement 3.2: Download Document 14](#_Toc370600452)

[System Feature 4.0: Globe Management 15](#_Toc370600453)

[Functional Requirement 4.1: Create Globe 15](#_Toc370600454)

[Functional Requirement 4.2: Modify Globe 15](#_Toc370600455)

[Functional Requirement 4.2.1: Add Document 15](#_Toc370600456)

[Functional Requirement 4.2.2: Add Group 16](#_Toc370600457)

[Functional Requirement 4.2.3: Add User 16](#_Toc370600458)

[Functional Requirement 4.2.4: Modify Permissions 16](#_Toc370600459)

[Functional Requirement 4.2.5: Add Comments 17](#_Toc370600460)

[Functional Requirement 4.3: Archive Globe 17](#_Toc370600461)

[System Feature 5.0: Globe Access 18](#_Toc370600462)

[Functional Requirement 5.1: Open Globe 18](#_Toc370600463)

[Functional Requirement 5.2: Modify Document 18](#_Toc370600464)

[Functional Requirement 5.3: Commit Changes 19](#_Toc370600465)

[Functional Requirement 5.4: Add Comments 19](#_Toc370600466)

[System Feature 6.0: Reporting 20](#_Toc370600467)

[Functional Requirement 6.1: View History 20](#_Toc370600468)

[Functional Requirement 6.2: Generate Report 20](#_Toc370600469)

[Non-Functional Requirements 21](#_Toc370600470)

[Performance/Response time requirement 21](#_Toc370600471)

[Availability Requirement 21](#_Toc370600472)

[Recoverability & Reliability Requirement 21](#_Toc370600473)

[Security Requirement 21](#_Toc370600474)

[Maintainability & Re-usability Requirement 22](#_Toc370600475)

[Portability & Extendibility Requirement 22](#_Toc370600476)

[Graphic User Interface 23](#_Toc370600477)

[Landing Page & Registration 23](#_Toc370600478)

[New Globe & Modify Permissions 23](#_Toc370600479)

[Open Globe 23](#_Toc370600480)

[System Architecture 24](#_Toc370600481)

[Arduino 24](#_Toc370600482)

[Local PC 24](#_Toc370600483)

[Windows Service 24](#_Toc370600484)

[Management Software 24](#_Toc370600485)

[Server 24](#_Toc370600486)

[API 25](#_Toc370600487)

[Database 25](#_Toc370600488)

[System Evolution 26](#_Toc370600489)

[Operating Systems 26](#_Toc370600490)

[Mobility 26](#_Toc370600491)

[Tasklock 26](#_Toc370600492)

[NFC 26](#_Toc370600493)

[Appendix A: Glossary 27](#_Toc370600494)

[Appendix B: Use Case Diagram 28](#_Toc370600495)

# Introduction

This document is a deliverable for a final year project as part of a Honours Degree in computing with the National College of Ireland.

A section titled ‘Overall Description’ will discuss the product perspective, an overview of the main features and the operating environment as well as any constraints on the projects development.

The System features section of the document will define a high level use case diagram which will encapsulate the main features of 'Globlock' and also drill down to the individual use case descriptions and functional requirements. Non-functional requirements are also listed here and mostly deal with the limits and efficiency that the system requires.

Firstly the document will outline the purpose, intended audience and scope of the requirements specification.

## Purpose

The purpose of this document is to define the Software Requirements specification for 'Globlock'.

'Globlock' is a combination of software and hardware that allows a high level user to manage a revision repository of documents and files while maintaining strict concurrency controls over these documents and files. The requirements laid out in this document will identify the functional and non-functional requirements for 'Globlock' as well as its operating environment, and development requirements or constraints. This section will also describe how the issue of user documentation and instruction will be tackled.

This document will allow the 'Globlock' project to be developed and implemented in the allocated timeline.

## Intended Audience

The intended audience for this document includes but is not limited to, the lecturers and faculty members of the National College of Ireland (listed in 'Distribution List' above) as well as external examiners tasked with marking the project.

The document is also intended for potential customers or users of the software and hardware elements of the product, upon project competition, and will also be made available to potential employers as an example of my work, capabilities and skills.

## Project Scope

The purpose of 'Globlock' (as stated in the Project Proposal) is to provide a means for unskilled or Professionals untrained in I.T. to gain access to and utilise the benefits of a repository management system.

'Globlock' provides a mechanism to tokenise files and projects, and allow a physical abstraction of these digital objects. This physical abstraction will allow strict concurrency control over the associated files/projects.

The software and hardware interfaces required for this abstraction and control are all encapsulated in 'Globlock'. It will provide the necessary interfaces to allow a high level user associate these files or projects with a physical object, as well as review and maintain a repository of versions, review changes, re-assign files or projects to different physical objects, and report on progress and updates from the 'Globlock' software.

It is important to note that 'Globlock' is not a replacement for existing repository systems and services. It is intended for use in environments where knowledge and training in concept and use of repository systems is unavailable. In particular it is directed at small to medium sized offices where concurrency control and revision reposition is needed or would add value to the business processes.

# Overall Description

## Product Perspective

The ‘Globlock’ software is a standalone set of applications, services and hardware components. It will provide the necessary tools required to allow technical and non-technical personnel make use of the benefits of a document repository, as well as allowing these documents and the permissions of their users, to be easily managed.

The operating environment will be covered in greater detail later in the document but in summary it will be windows based environments that could benefit from document control and repository management.

The ‘Globlock’ system will be created to allow a great deal of flexibility around the types of files that can be used, but at the same time provide a facility for a high level user to strictly control these files and their versions and permissions of the users that access them.

Great care will also be taken in development and design of ‘Globlock’, to remove the transparency of the architecture and allow the focus to be on the abstraction of the physical object representing digital files and documents.

Repositories are often difficult to use and require a level of training that is not always possible. The simplicity of its design and implementation will provide the fundamental benefits of a repository, without the complication of the steps and technical knowledge needed.

## Product Features

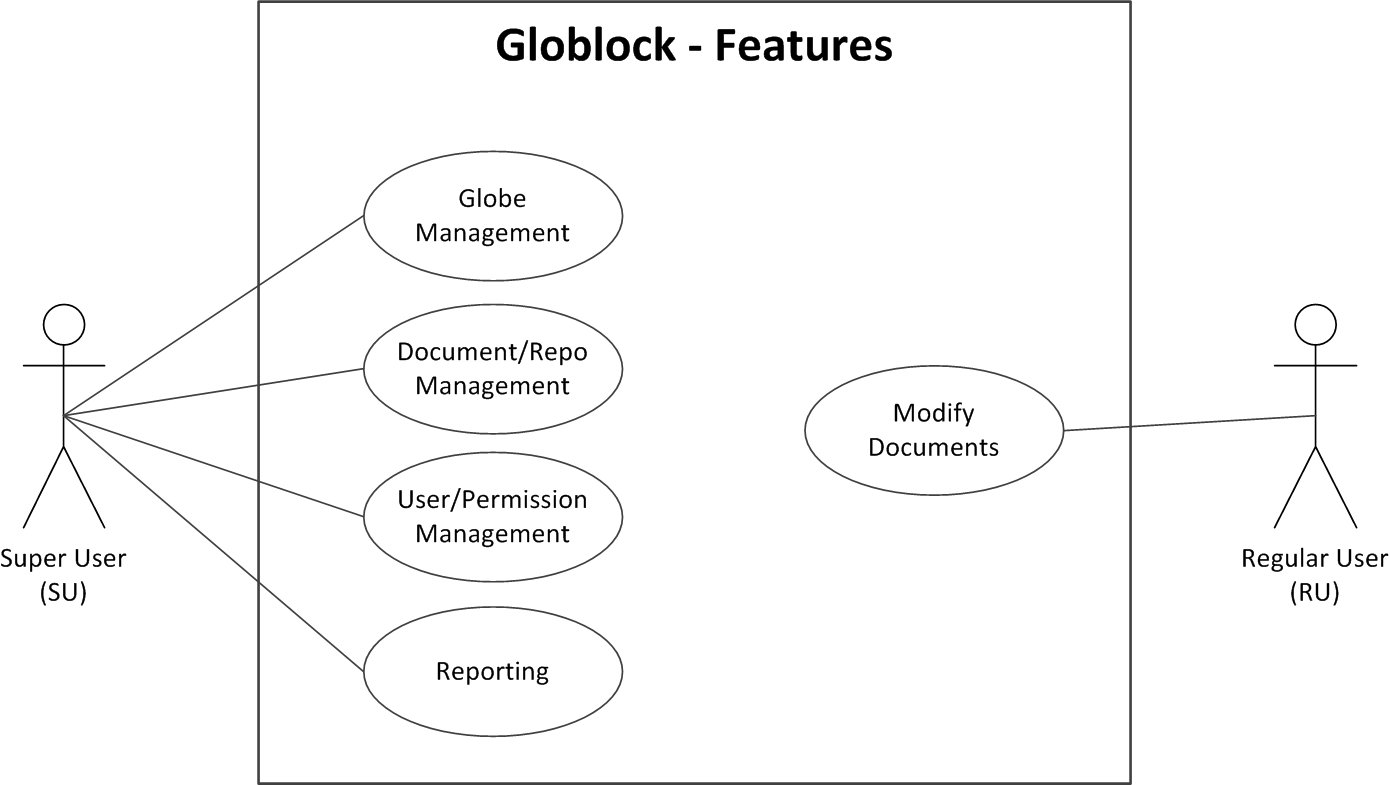


Figure 1.1

### User Account Management and Permissions

A very important feature of ‘Globlock’ will be the ability to manage user accounts. User classes will be covered in greater detail below, but a high level user will essentially control other user accounts and permissions over globes created and modified within the system.

A basic user will exist which will interact with the globes and make changes to the associated files, and comment on these changes as needed. These basic users will be restricted from accessing globes that they have not been given privilege to by the high level user.

Privileges of read or write may be assigned to each user on each globe.

### Repository and Document Management

As each globe is manipulated with by basic users and changes made to its associated files and documents, each iteration or revision of these will be stored and recorded along with any comments added. A central repository, the transparency of which will be hidden from the user, will store each revision or version. This in essence will provide the archiving functionality and management of documents.

If required, a high level user will have the facility to take a revision or version of a single document, or entire group of documents, and extract from the central archived repository and associate with a new or existing Globe.

### Globe Management

Each Globe will be managed by high level users of the system. A function of the software will allow the high level users to assign individual as well as groups of users access rights to the Globes created. These permissions will be basic read/write but will allow users to be assigned to Globes for information only. Users without write permissions will not be able to update or make changes to revisions of the Globe or its associated files or documents.

### Reporting

Another feature of the system will be basic reporting functionality which will allow high level users to track changes to Globes, files and documents and also track what users have interacted with the Globes.

## User Classes, Characteristics and Perspective

### Super User

The SU or high level user will be act as an administrator for other regular users. They will be responsible for the account creation and management. They will be tasked with creating users and assigning them to user groups if needed.

They will be solely responsible for creating new Globes and assigning files and documents to them. They have the ability to add comments to existing Globes for visibility to users.

Only SU’s will have the ability to retrieve previous revisions or versions from the repository. The reporting functionality also mentioned earlier will only be available to a SU.

### Regular user

A RU or regular/basic user will interact with the system more often than a SU but have the least amount of capability and control over Globes and associated files.

They will be assigned Globe privileges by the SU and if given write privileges, will be able to interact with and update documents which will in turn create new versions/iterations and assign older versions/iterations to the archive/repository. In essence their visibility to the functionality of the system will be the modification of documents and files.

At each iteration and prior to committing changes, the RU will be able to add comments, which may be used to document the changes made.

## Operating Environment

The operating environment for ‘Globlock’ will be based on a Windows environment, however future scope covered later in the document will describe in more detail that the system has been designed and development with extendibility and portability in mind.

### Globes

The Globes with embedded RFID will be of durable soft sponge material. The Globes themselves are readily available as are the RFID’s that will be embedded. The form factor of the RFID is yet to be determined as different types may need to be tested before a decision is made.

### Block Device

An Arduino device will be used to read the Globes embedded RFID tag. The device will require ‘Arduino sketches’ to be loaded onto the reader device to perform this function. The language needed for this has yet to be determined but will likely be in the C++/C programming language.

The device will be located on a desk or table, normally associated with an office environment and connected to the client PC via USB or Ethernet. This will allow communication to and from the client PC (such as the embedded id).

### Windows Service and Operating System

The windows service that will run in order to facilitate communication from the Block to the client PC will be based on x86 architecture. The architecture in mind during development will be Windows 7 and Windows 8. Home, Professional and Enterprise editions of this operating system will be thoroughly tested at both prototype and final stages.

### Management Client Software

The client software used by the SU to create and modify the account, Globes and other users of the system, will be accessible from the client PC but a platform and language decision for the management suite has yet to be determined.

### Web Server & Database

The webserver will be hosted remotely and the client PC will connect to it via the internet. The webserver will host a landing page for ‘Globlock’ and also an installation of MySQL which will act as the central database for the system.

### Application Programming Interface

The API will reside on the web server and provide responses to request such as session and authentication token information for the client PC.

### NFC Device

An Android NFC capable device will be used to act as a Block in the absence of an Arduino device. This NFC capable smart phone will provide the necessary portability of the system.

## Design and Implementation Constraints

The design of the system must allow for low end generic PC builds to interact and utilise the components of Globlock with negative effects.

However, due to the file/document transfer features of Globlock, the client PC’s must have a minimum level of broadband to allow the system to function without risk of dropout.

The design and architecture allows for implementation in Windows based operating environments however future scope (discussed later) will allow for multiple platforms and architectures.

Transfer protocols to be used for files and document transfer to and from the central store have yet to be determined.

The system must be secure enough to prevent unwarranted access to the files and documents uploaded and downloaded to and from the system so a method of encryption will be required during transport of data. This has yet to be determined.

## User Documentation

On completion of the prototype, an instructional manual as well as online tutorial demonstrating the different features will be available and hosted on the landing website.

The entire system will be designed with the user experience in mind so ease of use and intuitive application behaviour will be high on the agenda.

## Definitions, Acronyms and Abbreviations

See Appendix A: Glossary

# System Features

See Figure 1.1 for a high level overview of the features of Globlock.

## System Feature 1.0: Account Management

Account management allows a SU to register an account on the system to avail of the Globlock services and functionality. After account creation the SU must login for the first time. All types of users are required to login to use the system if a session does not already exist.

### Functional Requirement 1.1: Create Account

|  |  |
| --- | --- |
| Use Case: | Create Account |
| Scope: | SU creates an account to avail of the features of the system. |
| Description: | In order to use the system, the user must previously have registered and created an account. It is also possible to use the system as a default user as long as a SU has created an associated user account. |
| Actors: | SU. |
| Assumptions: | System is available, user has the necessary browser and internet connection is active. User has an active and valid email address. |
| Pre-Conditions: | SU has not been previously registered on the system. |
| Activation: | The use case starts when the SU navigates to the 'Globlock Website'. |
| Main Flow: | 1. SU navigates to 'Globlock' website. 2. User selects ‘Create New Account’. 3. User is asked to enter the required details. 4. Details are verified by the system against the DB and verified to be unique. (I.E. the account does not already exist.) 5. All details are validated and stored in the DB. 6. Verification link is sent to the email account specified by the user. 7. User follows the link and verifies account. |
| Exceptional Flow 1: | 1. User Cancels registration process at any stage. 2. Account details are removed from the system. |
| Exceptional Flow 2: | 1. User does not respond to account verification email within 24 hours. 2. Account details are removed from the system. |
| Post Condition: | Account is created and exists on the system. |

### Functional Requirement 1.2: Login

|  |  |
| --- | --- |
| Use Case: | Login |
| Scope: | SU creates an account to avail of the features of the system. |
| Description: | In order to use the system the SU must previously have registered and created an account. In the case of a RU, an SU must have previously created a user account. |
| Actors: | SU, RU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally. |
| Pre-Conditions: | SU has been previously registered on the system.  RU has had account created previously by SU. |
| Activation: | The use case starts when the SU/RU navigates to the 'Globlock’ Website, or opens the ‘Globlock’ software locally. |
| Main Flow: | 1. User navigates to 'Globlock' website. 2. User selects ‘Login’. 3. User is asked to enter the required details. 4. Details are verified by the system against the DB. 5. User session in created. |
| Alternate Flow: | 1. User starts the 'Globlock' software. 2. Go to Step 2. |
| Exceptional Flow 1: | 1. User Cancels login process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. User details are incorrect. 2. User is re-directed back to main page. |
| Post Condition: | User is logged into system and user session in created. |

## System Feature 2.0: User Management

A function of the system is the facility to allow a SU to create RU accounts so that they may interact with the system.

### Functional Requirement 2.1 Create Group

|  |  |
| --- | --- |
| Use Case: | Create Group |
| Scope: | SU creates a User Group. |
| Description: | User Group is created by SU to allow multiple users to be grouped for access to ‘Globes’. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has Logged into the system. (UC 1.2.) |
| Activation: | The use case starts when the SU selects ‘Create User Group’. |
| Main Flow: | 1. SU selects ‘Create User Group’. 2. SU is asked to enter the required details. 3. Details are verified by the system against the DB. 4. User Group is created. |
| Exceptional Flow 1: | 1. User Cancels ‘User Group’ creation process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. User Group details are invalid. 2. User is asked to re-enter details and try again. 3. Go to Step 3. |
| Post Condition: | User Group is created on the system. |

### Functional Requirement 2.2: Create User

|  |  |
| --- | --- |
| Use Case: | Create User |
| Scope: | SU creates a User. |
| Description: | RU User account is created by SU to allow users access to ‘Globes’, and to be assigned to groups. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has Logged into the system. (UC 1.2.) |
| Activation: | The use case starts when the SU selects ‘Create User’. |
| Main Flow: | 1. SU selects ‘Create User Group’. 2. SU is asked to enter the required details. 3. Details are verified by the system against the DB. 4. User Account is created. |
| Exceptional Flow 1: | 1. User Cancels ‘User Account’ creation process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. User details are invalid. 2. SU is asked to re-enter details and try again. 3. Go to Step 3. |
| Post Condition: | User Account is created on the system. |

### 

### Functional Requirement 2.3: Modify Group

|  |  |
| --- | --- |
| Use Case: | Create Group |
| Scope: | SU creates a User Group. |
| Description: | Modification of previously created User Groups. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has Logged into the system. (UC 1.2.) |
| Activation: | The use case starts when the SU selects ‘Modify User Group’. |
| Main Flow: | 1. SU selects ‘Modify User Group’. 2. User modifies the required fields and details. 3. Details are verified by the system against the DB. 4. User Group is updated. |
| Exceptional Flow 1: | 1. User Cancels ‘User Group’ modification process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. User Group details are invalid. 2. User is asked to re-enter details and try again. 3. Go to Step 3. |
| Post Condition: | User Group is updated on the system. |

### Functional Requirement 2.4: Modify User

|  |  |
| --- | --- |
| Use Case: | Modify User |
| Scope: | SU modifies a User Account. |
| Description: | Modification of previously created User Groups. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has Logged into the system. (UC 1.2.) |
| Activation: | The use case starts when the SU selects ‘Modify User’. |
| Main Flow: | 1. SU selects ‘Modify User’. 2. User modifies the required fields and details. 3. Details are verified by the system against the DB. 4. User Account is updated. |
| Exceptional Flow 1: | 1. User Cancels ‘Modify User’ process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. User details are invalid. 2. SU is asked to re-enter details and try again. 3. Go to Step 3. |
| Post Condition: | User Account is updated on the system. |

## System Feature 3.0 Document Management

As the goal of the system is to provide a repository and archive for document management, the ability to upload and download documents is a functional requirement.

### Functional Requirement 3.1: Upload Document

|  |  |
| --- | --- |
| Use Case: | Upload Document |
| Scope: | SU uploads a Document for use on the system. |
| Description: | In order to manage documents through the use of Globes, the documents must be first uploaded onto the system. |
| Actors: | SU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally. |
| Pre-Conditions: | SU has Logged into the system. (UC 1.2.) |
| Activation: | The use case starts when the SU selects ‘Upload Document’. |
| Main Flow: | 1. SU selects ‘Upload Document’. 2. REPEAT    1. User selects Document on Local PC.    2. Details are verified by the system against the DB.   UNTIL no more documents required for upload   1. Document(s) is uploaded to the system. |
| Exceptional Flow 1: | 1. User Cancels upload process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. Document details are invalid. 2. User is re-directed back to main page. |
| Post Condition: | Document(s) is uploaded and exists on the system. |

### Functional Requirement 3.2: Download Document

|  |  |
| --- | --- |
| Use Case: | Download Document |
| Scope: | SU downloads a document currently in use on the system. |
| Description: | Documents may or may not have had changes made to them since initial upload. The SU will be asked what version is required for download. |
| Actors: | SU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally. |
| Pre-Conditions: | SU has Logged into the system. (UC 1.2.) |
| Activation: | The use case starts when the SU selects ‘Download Document’. |
| Main Flow: | 1. SU selects ‘Download Document’. 2. User Selects document from the document list. 3. User is prompted for location to download document to. 4. Document is downloaded from the system. |
| Alternate Flow: | 1. Alternate or multiple versions of the Document exist 2. User is prompted for which version of the software to download 3. Go to Step 3. |
| Exceptional Flow 2: | 1. Download location is invalid. 2. User is re-directed back to main page. |
| Post Condition: | Document(s) is downloaded and exists on the local PC. |

## System Feature 4.0: Globe Management

Globe management is an encapsulation of all the functional requirements that will be needed for documents to be tokenised and accessible from a RU or group of users.

### Functional Requirement 4.1: Create Globe

|  |  |
| --- | --- |
| Use Case: | Create Globe |
| Scope: | SU creates a new Globe. |
| Description: | Globes are created to allow documents to be associated and provide concurrency control over these documents. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has Logged into the system (UC 1.2.). Device is connected to the system to read embedded RFID from the physical Globe. |
| Activation: | The use case starts when the SU selects ‘Create New Globe’. |
| Main Flow: | 1. SU selects ‘Create New Globe’. 2. SU is asked to enter the required details. 3. Details are verified by the system against the DB. 4. User is asked to place physical Globe on the block device. 5. Id from embedded device is read and logged in the system. 6. Globe is associated with the Physical Globe. 7. SU is asked to remove the physical Globe. 8. Globe is created and active on the system. |
| Exceptional Flow 1: | 1. User Cancels ‘Create Globe’ process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. Physical ID is associated with another Globe. 2. User is asked to remove Globe and try again. 3. Go to Step 5. |
| Post Condition: | Globe is created and active on the system. |

### Functional Requirement 4.2: Modify Globe

### Functional Requirement 4.2.1: Add Document

|  |  |
| --- | --- |
| Use Case: | Add Document |
| Scope: | SU adds a Document to a Globe. |
| Description: | In order to manage documents through the use of Globes, the documents must be first uploaded onto the system. Once uploaded and a Globe is created, they may be assigned to a Globe for update from RU. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has created a Globe on the system. (UC4.2)  Globe has been created and is available for modification on the system. |
| Activation: | The use case starts when the SU selects ‘Add Document to Globe’. |
| Main Flow: | 1. SU selects ‘Add Document to Globe’. 2. REPEAT    1. User selects Document from document list.    2. Details are verified by the system against the DB.   UNTIL no more documents required for addition   1. Document(s) is added to the Globe on the system. |
| Alternate Flow: | 1. Alternate or multiple versions of the Document exist 2. User is prompted for which version of the software to download 3. Go to Step 3. |
| Exceptional Flow 1: | 1. User Cancels process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. Document details are invalid. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. Document is associated with another Globe. 2. User is re-directed back to main page. |
| Post Condition: | Document(s) is added to the Globe on the system. |

### Functional Requirement 4.2.2: Add Group

|  |  |
| --- | --- |
| Use Case: | Add Group |
| Scope: | SU assigns a group of users to the Globe. |
| Description: | In order for users to access documents and use the Globe through the system the SU must previously have assigned them to the Globe they wish to access. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has created a Globe on the system. (UC4.2)  Globe has been created and is available for modification on the system. |
| Activation: | The use case starts when the SU selects ‘Add Group to Globe’. |
| Main Flow: | 1. SU selects ‘Add Group to Globe’. 2. REPEAT    1. User selects User Group from Group list.    2. Details are verified by the system against the DB.   UNTIL no more Groups required for addition   1. Group (s) is assigned to the Globe on the system. 2. Group is assigned Read Only permission. |
| Exceptional Flow 1: | 1. SU Cancels process at any stage. 2. SU is re-directed back to main page. |
| Post Condition: | User Group(s) are assigned to the Globe on the system |

### Functional Requirement 4.2.3: Add User

|  |  |
| --- | --- |
| Use Case: | Add User |
| Scope: | SU assigns a User to the Globe. |
| Description: | In order for users to access documents and use the Globe through the system the SU must previously have assigned them to the Globe they wish to access. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has created a Globe on the system. (UC4.2)  Globe has been created and is available for modification on the system. |
| Activation: | The use case starts when the SU selects ‘Add User to Globe’. |
| Main Flow: | 1. SU selects ‘Add User to Globe’. 2. REPEAT    1. User selects User from User list.    2. Details are verified by the system against the DB.   UNTIL no more Users required for addition   1. User(s) are assigned to the Globe on the system. 2. User is assigned Read Only permission. |
| Exceptional Flow 1: | 1. SU Cancels process at any stage. 2. SU is re-directed back to main page. |
| Post Condition: | User Group(s) are assigned to the Globe on the system |

### Functional Requirement 4.2.4: Modify Permissions

|  |  |
| --- | --- |
| Use Case: | Modify Permissions |
| Scope: | SU modifies the permissions for a User or Group associated with a Globe. |
| Description: | To use the Globe through the system users have been previously assigned to a Globe they wish to access, and assigned privileges. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has created a Globe on the system. (UC4.1)  Globe has been created and is available for modification on the system.  Groups (UC4.1.2) or Users (UC4.1.3) have been added to the Globe. |
| Activation: | The use case starts when the SU selects ‘Modify Permissions’. |
| Main Flow: | 1. SU selects ‘Add User to Globe’. 2. REPEAT    1. SU selects User from User list in Globe.    2. User modifies privilege   UNTIL no more Users required for addition   1. User(s) are assigned to the Globe on the system. |
| Alternate Flow 1: | * 1. SU selects Group from User list in Globe. |
| Post Condition: | User and/or Group privileges have been modified on the Globes. |

### Functional Requirement 4.2.5: Add Comments

|  |  |
| --- | --- |
| Use Case: | Add Comments |
| Scope: | SU adds comments to a Globe for viewing by the next RU to access the Globe. |
| Description: | To allow SU to add comments to a Globe currently active on the system which are then visible to the next RU. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has created a Globe on the system. (UC4.1)  Globe has been created and is available for modification on the system.  Groups (UC4.1.2) or Users (UC4.1.3) have been added to the Globe. |
| Activation: | The use case starts when the SU selects ‘Add Comments’. |
| Main Flow: | 1. SU selects ‘Add Comments to Globe’. 2. User enters appropriate comments in the fields provided. 3. SU comments are assigned to the Globe on the system. |
| Exceptional Flow 1: | 1. SU Cancels process at any stage. 2. SU is re-directed back to main page. |
| Post Condition: | Comments are assigned to the Globe on the system. |

### Functional Requirement 4.3: Archive Globe

|  |  |
| --- | --- |
| Use Case: | Archive Globe |
| Scope: | Remove associations with physical Globe, remove users and archive all associated documents. |
| Description: | Globes are created to allow documents to be associated and provide concurrency control over these documents. On completion of all changes the Globe may be committed to archive by the SU.  Documents previously associated with a Globe may now be associated with a separate physical Globe. |
| Actors: | SU |
| Assumptions: | System is available and user has started the necessary software. |
| Pre-Conditions: | SU has Logged into the system (UC 1.2.). Device is connected to the system to read embedded RFID from the physical Globe. |
| Activation: | The use case starts when the SU places the Globe on the Block. |
| Main Flow: | 1. SU places Physical Globe on Block. 2. System verifies ID is associated with Active Globe on the system. 3. SU selects ‘Archive Globe’. 4. SU is asked to enter comments relating to the archive. 5. Globe is disassociated with the Physical Globe. 6. All associated documents are removed from the Globe on the system. 7. Globe is marked as Archived. 8. SU is asked to remove the physical Globe. 9. Globe is now archived on the system. |
| Exceptional Flow 1: | 1. User Cancels ‘Archive Globe’ process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. Physical ID is not associated with a Globe on the system. 2. User is asked to remove Globe. 3. SU is re-directed back to main page. |
| Post Condition: | Globe is now archived on the system. |

## System Feature 5.0: Globe Access

After Globes have been created, documents uploaded and associated and users assigned permissions, there then exists a requirement for users to open the newly created Globes and modify their contents.

### Functional Requirement 5.1: Open Globe

|  |  |
| --- | --- |
| Use Case: | Open Globe |
| Scope: | RU accesses a Globe. |
| Description: | In order to make changes to document the RU accesses the Globe by placing it on the block. |
| Actors: | RU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally.  User has the necessary Block device connected to the PC. |
| Pre-Conditions: | RU has had an account created previously by SU (UC2.2).  RU has logged into the system (UC1.2). |
| Activation: | The use case starts when the RU places the Globe on the Block. |
| Main Flow: | 1. User places the Globe on the Block. 2. RFID is read by the windows service. 3. Details are verified by the system against the DB. 4. Associated Globe documents are downloaded locally and opened by the software. |
| Exceptional Flow 1: | 1. User Cancels process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. User has not been assigned to the Globe. 2. User is re-directed back to main page. |
| Post Condition: | Globe documents are downloaded locally and opened by the software |

### Functional Requirement 5.2: Modify Document

|  |  |
| --- | --- |
| Use Case: | Open Globe |
| Scope: | RU accesses a Globe and makes changes to the documents associated with it. |
| Description: | In order to make changes to the document the RU accesses the Globe by placing it on the block. The documents are then downloaded locally and opened in their default software. User can now modify these documents as they wish. |
| Actors: | RU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally.  User has the necessary Block device connected to the PC. |
| Pre-Conditions: | RU has opened the Globe (UC5.1) |
| Activation: | The use case starts when the documents are opened. |
| Main Flow: | 1. Documents are opened by the System. 2. Documents are modified by the RU. 3. Documents downloaded are compared against the repository. 4. Documents are flagged as having differences. |
| Exceptional Flow 1: | 1. User makes no change to the documents. |
| Exceptional Flow 2: | 1. User does not have write privileges over the documents. |
| Post Condition: | Documents are modified and flagged as having differences on the system. |

### Functional Requirement 5.3: Commit Changes

|  |  |
| --- | --- |
| Use Case: | Commit Changes |
| Scope: | RU commits changes to the documents. |
| Description: | The user may decide to commit the changes made to the repository. There are 2 ways of completing this. By selecting this option in the software or by removing the Globe from the Block. |
| Actors: | RU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally.  User has the necessary Block device connected to the PC. |
| Pre-Conditions: | RU has opened the Globe (UC5.1) and modified it (UC5.2) |
| Activation: | The use case starts when the Globe is removed or the Commit option is selected from the software. |
| Main Flow: | 1. User removes the Globe from the Block. 2. User is prompted if they want to commit the changes to the Globe. 3. User selects yes. 4. Documents versions are iterated in the system. 5. Documents are uploaded to the central repository. |
| Alternate Flow: | 1. User selects to ‘Commit changes’ from the software. 2. Go to Step 2 above. |
| Exceptional Flow: | 1. User aborts the commit operation. 2. Files are removed from the local PC and no changes made to the central repository. |
| Post Condition: | 1. Documents are uploaded to the central repository. |

### Functional Requirement 5.4: Add Comments

|  |  |
| --- | --- |
| Use Case: | Add Comments |
| Scope: | Add Comments to the recent Commit. |
| Description: | The user may decide to commit the changes made to the repository and if so they are provided with a facility to optionally add comments to the recent commit. |
| Actors: | RU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally.  User has the necessary Block device connected to the PC. |
| Pre-Conditions: | RU has opened the Globe (UC5.1), modified it (UC5.2) and Committed changes. |
| Activation: | The use case starts when Globe has been committed successfully. |
| Main Flow: | 1. Globe is committed successfully. 2. RU is prompted to add comments. 3. RU enters comments in the appropriate field. 4. Comments are added to the Globe on the system. |
| Exceptional Flow 1: | 1. RU aborts the process at any stage. 2. User is re-directed back to main page. |
| Exceptional Flow 2: | 1. RU declines the prompt to add comments. 2. User is re-directed back to main page. |
| Post Condition: | Documents are uploaded to the central repository. |

## System Feature 6.0: Reporting

A very basic functionality from the system is the ability to report on current and archived Globes that have been created, and the changes that have occurred to them.

### Functional Requirement 6.1: View History

|  |  |
| --- | --- |
| Use Case: | View History |
| Scope: | View history for a particular Globe. |
| Description: | Simple reporting functionality will allow a SU to view the history of changes and commits made to a Globe. |
| Actors: | SU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally. |
| Pre-Conditions: | SU has created a Globe on the system (UC4.2).  Globe has been created and is available for modification on the system.  Globe has been modified and changes have been committed (UC5.2). |
| Activation: | The use case starts when the SU selects View History. |
| Main Flow: | 1. User selects the required ‘Globe’ from active Globes on system. 2. User selects ‘View History’. 3. Details are retrieved by the system from the DB. 4. Historical details of the Globe are displayed to the SU. |
| Exceptional Flow: | 1. User Cancels process at any stage. 2. User is re-directed back to main page. |
| Post Condition: | Historical details of the Globe are displayed to the SU. |

### Functional Requirement 6.2: Generate Report

|  |  |
| --- | --- |
| Use Case: | Generate Report |
| Scope: | Generate Report on a particular Globe. |
| Description: | Simple reporting functionality will allow a SU to view the history of changes and commits made to a Globe, and generate a report of these historical changes. |
| Actors: | SU |
| Assumptions: | System is available and user has necessary browser and internet connection.  User has the necessary software installed locally. |
| Pre-Conditions: | SU has created a Globe on the system (UC4.2).  Globe has been created and is available for modification on the system.  Globe has been modified and changes have been committed (UC5.2).  User has selected View History for a particular Globe (UC6.1). |
| Activation: | The use case starts when the SU selects Generate Report. |
| Main Flow: | 1. User selects the required ‘Globe’ from active Globes on system. 2. Details are retrieved by the system from the DB. 3. Historical details of the Globe are displayed to the SU. |
| Exceptional Flow: | 1. User Cancels process at any stage. 2. User is re-directed back to main page. |
| Post Condition: | Historical details of the Globe are displayed to the SU. |

## Non-Functional Requirements

### Performance/Response time requirement

|  |  |
| --- | --- |
| Windows Service: | Windows service must not use more than 5% of system resources during idle periods and no more than 75% during non-idle for more than 15 seconds. This is to prevent the Globlock system being viewed as a burden or hindrance to the user.  This will be tested thoroughly during prototype and final stage testing. |
| Server Response: | Response from the server must not exceed 60 seconds for requests for primitive data queries. Download times should not exceed 5 minutes per document.  These requirements will be tested thoroughly during prototype and final stage testing. |
| Block Response: | It should not take longer 10 seconds to read Embedded RFID in Globes from the Block device. |

### Availability Requirement

|  |  |
| --- | --- |
| Server Availability: | Server must have minimum of 99% uptime. This will allow 6.72 hours over a four week period to allow for scheduled changes or unscheduled down time. |
| If server is unreachable. Documents must be stored locally until such time that the server becomes available. |

### Recoverability & Reliability Requirement

|  |  |
| --- | --- |
| Globe Loss: | As a Globe is a physical object it may become lost. A facility to Archive Globes already exists and will allow previous documents associated with a lost Globe to be assigned to a new Globe. |
| Software Error response: | All errors should be captured and in the event the system cannot commit changes the user will be clearly and effectively informed. |
| Hardware Failure: | In the event of hardware failure the management application should continue to allow the management of users and documents, until such time that the Globe can be docked, loaded and undocked. |

### Security Requirement

|  |  |
| --- | --- |
| User Access: | Users without previously assigned access should not be able to access the files. Users with only read privileges should not be able to write or make changes. |
| User accounts, both super (SU) and regular (RU) will require passwords. Passwords complexity will be defined the ISO27001 standard for Information Security. |
| Communication Transport: | A method of encryption (TBD) will be used to encrypt data packets and files while being transported to prevent breaches. |

### Maintainability & Re-usability Requirement

|  |  |
| --- | --- |
| Maintenance: | Updates to platform of systems should be easy to manage and maintain with no additional training by the users. |
| Re-Usability: | The Globes should be interchangeable and re-usable in that a super user may decide to assign new or archived information to a Globe, to prevent making the Globe objects obsolete. |
| Code Re-use: | Code shall be written in such a way as too allow as much flexibility as possible around it’s re-use. |

### Portability & Extendibility Requirement

|  |  |
| --- | --- |
| Mobility: | The block device should be used at a desktop location. In the event that a mobile solution is required and NFC capable smart phone shall be used in place of the block. |
| The system will be designed and Developed to allow the use of other operating systems and devices in future context. |
| The system will be designed and developed to allow flexibility for future scope additions. An example of this is the ability to associate a Globe placed on the NFC device or Block device with particular tasks (see System Evolution\TASKLOCK). |

# Graphic User Interface

Below are some mock-ups of the types of simple interfaces to be expected from the management Application. An intuitive and easy to use design will be the goal of the user interface.

## Landing Page & Registration

|  |  |
| --- | --- |
| Figure 2.1 | Figure 2.2 |

## New Globe & Modify Permissions

|  |  |
| --- | --- |
| Figure 2.3 | Figure 2.4 |

## Open Globe

|  |  |
| --- | --- |
| Figure 2.5 | Figure 2.6 |

# System Architecture

The system architecture of the project can be divided into 3 main sections; the Arduino/NFC device, the local PC and finally the Web/DB server. An overview of the system can be seen in figure 3.1 below, with a very basic flow of data through the system.

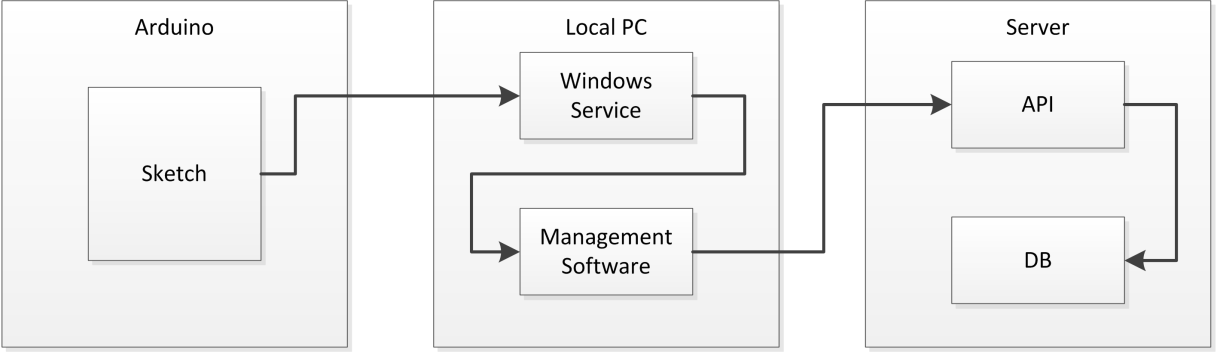


Figure 3.1

## Arduino

The Arduino device will require software to be loaded to the device in order for it to communicate with the windows service running on the client PC. The language for these embedded programs or sketches has yet to be determined.

## Local PC

The local or client PC will have 2 separate software components, a windows service and the management software suite.

### Windows Service

The Windows service which communicates with the Arduino device to read embedded RFID’s and also communicate with the server, will reside on the client PC and run continuously monitoring port communication from the Arduino.

The language for this windows service has yet to be determined.

### Management Software

The second software component on the client PC is the Management Software which will allow SU and RU interaction with the system.

The languages, API’s and frameworks used for the creation of the management software have yet to be determined, however as the operating environment requirement is for a Windows based environment, it is likely the these tools will be windows friendly.

## Server

The server that will be used as a web portal for the system will be hosted remotely, through a third party. There will be 2 components required for Globe modification and interaction and these are the API and the Database. Figure 3.2 gives an abstracted view of the system with regard to the remote server.

### API

The API to allow communications to and from the DB as well as the DB itself will be located on the DB server. This server will also act as a web server. In future version of the system where greater expansion is required these two components (web and DB) may be separated.

### Database

The database of choice will likely be a strain of SQL but has yet to be determined.

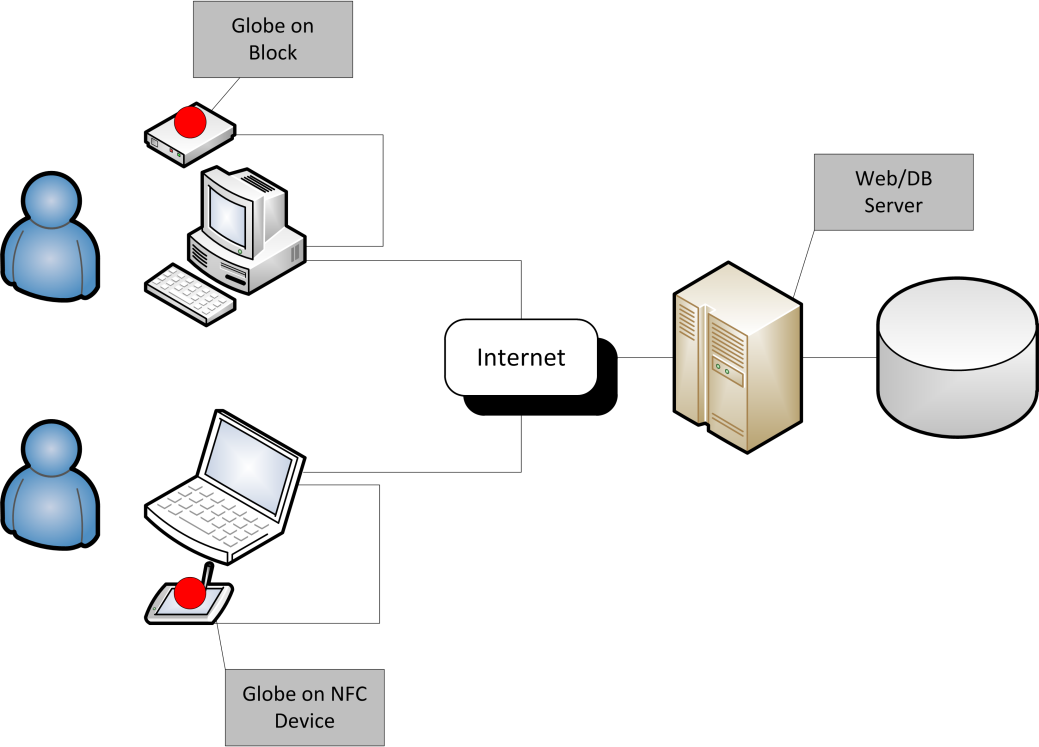


Figure 3.2

# System Evolution

The design and development of Globlock will allow for great flexibility around the systems evolution and expansion. The components of the system will be developed as individual units to allow for expansion, replacement and evolution.

## Operating Systems

Through iteration of the system it is hoped that other operating systems will be tested to allow cross platform usage and portability.

## Mobility

A greater focus on smart phones and tablets would allow for reduced hardware constraints and greater mobility for the system.

## Tasklock

The re-usability of the software when built in individual components will allow for divergence from the main concept, while maintain the abstraction of tokenization of files.

An example of this is using RFID tags placed on a desk to manage tasks. As an NFC capable device is passed over the tag, the associated task is displayed on the screen. As the user of the system completes tasks, they move to the next on the list by moving the device over the next RFID tag placed on the desk.

## NFC

According to NFC world (NFC World, 2013), 1 in 3 smartphones being produced have NFC capability.

# Appendix A: Glossary

|  |  |  |
| --- | --- | --- |
| **Abbreviation** | **Name** | **Description** |
| API | Application Programming Interface | Software that allows communication with other software or systems. |
| - | Block | Arduino device capable of reading embedded RFID. |
| DB | Database | Grouped Data. |
| SU | Super User | High level administrative user of ‘Globlock’. See User Classes, Characteristics and Perspective. |
| - | Globes | Small sponge objects with embedded RFID. |
| - | Globlock | The Globlock software, hardware and supporting systems. |
| I.T. | Information Technology | Application of computers and telecommunication equipment. |
| NFC | Near Field Communication | Wireless, non-contact radio frequency communication. |
| PC | Personal Computer | Desktop, Laptop or thin client. |
| RFID | Radio Frequency ID | Wireless, non-contact radio frequency identification. |
| RU | Regular User | Low level user of ‘Globlock’. See User Classes, Characteristics and Perspective. |
| TDB | To be determined | Currently undefined or yet to be determined. |

# Appendix B: Use Case Diagram

