**Mark Lese**

**Dustin Wolf**

**Jot Requirements**

**September 23, 2013**

**ACS560 – Dr. Chen**

|  |  |  |
| --- | --- | --- |
| **Change History** | | |
| **By** | **Date** | **Description** |
| M. Lese | 09/23/2013 | Initial |
|  |  |  |

**Table of Contents**

[1 Introduction 1](#_Toc367644165)

[2 Glossary 1](#_Toc367644166)

[3 Requirements 1](#_Toc367644167)

[3.1 Functional Requirements 1](#_Toc367644168)

[3.1.1 User Management 1](#_Toc367644169)

[3.1.2 Entity Management 1](#_Toc367644170)

[3.2 Non-functional Requirements 2](#_Toc367644171)

[3.2.1 System 2](#_Toc367644172)

[3.2.2 Encryption 2](#_Toc367644173)

[3.2.3 User Management 2](#_Toc367644174)

[4 System Architecture 3](#_Toc367644175)

[**Appendix A** 5](#_Toc367644176)

# Introduction

**Jot** is a media storage application to store images, videos, text files and other notes into a cloud database system and to help manage the various media. The idea of this is to allow users to take pictures of notes or diagrams, upload a document, send a quick message, video tape a quick snippet of a lecture, or even record an audio clip. **Jot** allows the user to send data to another registered user.

# Glossary

**Client** – A device, PC, phone, etc., that is used to communicate with the server.

**Cloud** – An abstract storage area available to user via various means.

**Data** – Any piece of information the user wishes to store on the cloud server. This data may be a text document or media.

**Entity** – A collection of data sent to or retrieved from the server.

**Media** – An image, video, or audio recording.

**Registered User** – A user that has successfully registered with the server.

**Server** – The system that holds the user’s data and communicates with the clients.

**Share** – When one user allows access to his data to some other registered user.

**Title** – A simple text given to an entity that is unique to other entities owned by the user.

# Requirements

This section outlines the functional and non-functional requirements of **Jot**. Each requirement shall be denoted by **[Rn]** where n is the requirement number. These requirement notations shall be used during validation to ensure all requirements are met.

## Functional Requirements

This section defines **Jot’s** provided services.

### User Management

[R1] **Jot** shall provide a user the means to register with the server using a username and password.

[R2] **Jot** shall allow a registered user to logon with a username and password prior to interaction with other parts of the system.

[R3] **Jot** shall allow a registered user to delete her account.

### Entity Management

A user in this section is assumed to be a logged-on user.

[R4] **Jot** shall provide a means for the user to select and gather data to group into an entity.

[R5] **Jot** shall allow a user to store an entity to the cloud keyed by a title.

[R6] **Jot** shall allow a user to share an entity or parts of an entity with another registered user.

[R7] **Jot** shall allow a user to retrieve and view his entity or parts of his entity.

[R8] **Jot** shall allow retrieval of media by type or storage date.

[R9] **Jot** shall allow a user to edit the entity, such as its annotation or contents.

[R10] **Jot** shall allow a user to retrieve and view other users’ shared entities.

[R11] **Jot** shall allow a user to delete an entity or parts of an entity.

## Non-functional Requirements

This section defines the constraints Jot shall adhere to.

### System

[R12] **Jot’s** server shall be a Linux-based server (see **Appendix A** at the end of this document).

[R13] The server-side software shall be written in PHP.

[R14] The server shall provide database storage for entities and user information.

[R15] The database shall be configured to allow for transactional operation.

[R16] The client application shall be available for a Windows PC, Android device, or iOS device.

[R17] The client shall be written in a language that is not PHP.

### Encryption

[R18] Since the system may handle personal data, **Jot** shall encrypt all communications between the client and server.

### User Management

[R19] Only registered users shall store or retrieve data from the server.

[R20] Registered users accounts shall be secured by a user-defined username and password.

[R21] User’s passwords shall be encrypted in the database.

[R22] Users shall be allowed to store no more than a maximum amount of data per his account; configurable within the system.

# System Architecture

This section shows an expected high-level system diagram in **Figure 4-1**. The client side shows the Android and iOS devices along with a Windows PC. The server side shows the PHP web services hosted on a Linux OS. The server also shows the likely MySQL database for storage.

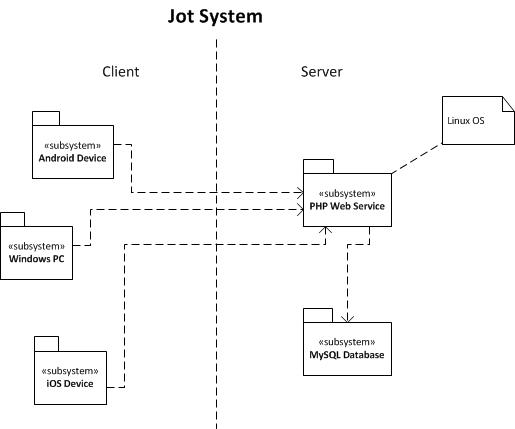


Figure 4-1

On the following page, **Figure 4‑2** shows a client prototype (subject to change) of the initial page presented to the user.

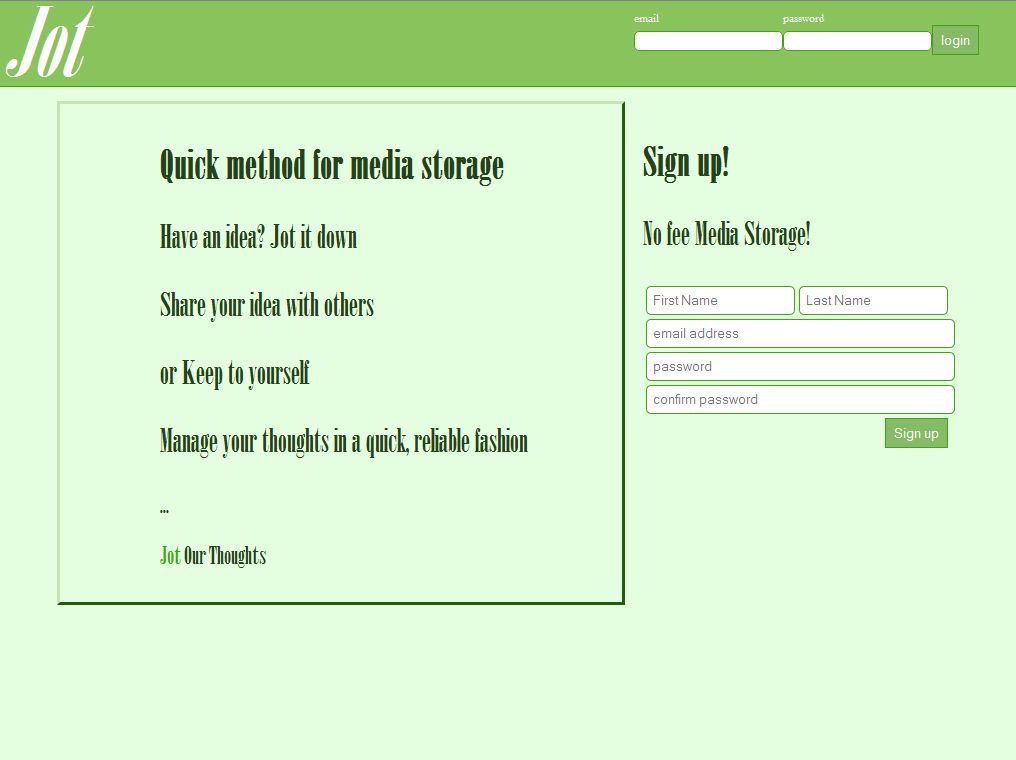


Figure 4-1

**Appendix A**

**Figure A-1** shows a Raspberry Pi module. This device, chosen for its novelty, is easily configured to run Debian Linux OS. The OS and all applications reside on a 16GB SD memory card. This is of sufficient size for the class project, but not practical for a “full production” system due to its limited memory.

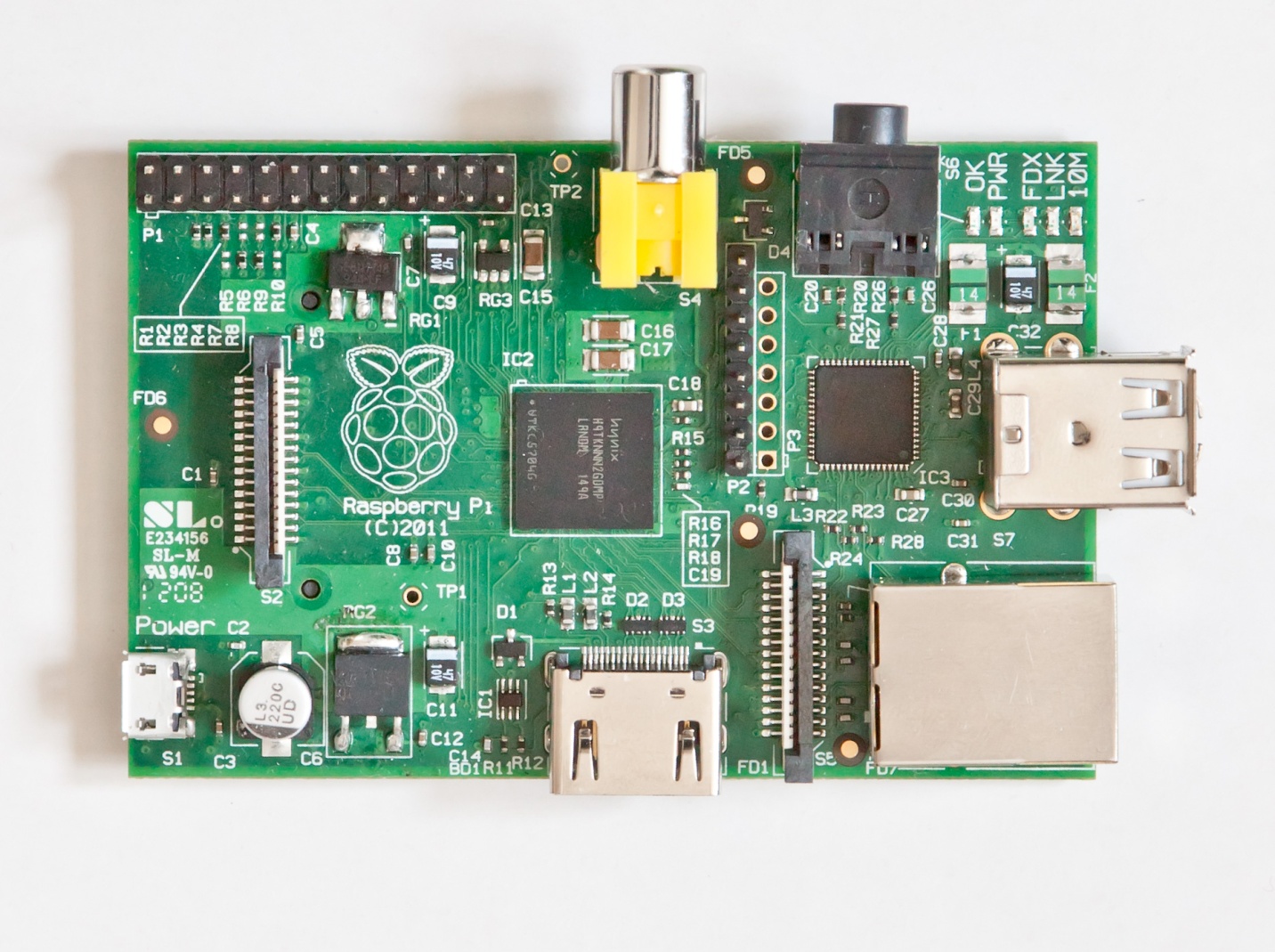


Figure A-1