
SYSTEM DESIGN & QUERIES



Mobile App Team

LinkMeEasy App

12 June 2019

TABLE OF CONTENTS

Revision History	4
queries	5
Purpose	Errore. Il segnalibro non è definito.
Abbreviation List	6
Roles and Responsibilities	7
Project References	7
Design Considerations	7
Assumptions	7
Constraints	7
Contingency	7
Strategies	7
Technical Architecture	8
System Overview	8
Logical View	9
Component View	9
Component Descriptions	9
Technical Requirements	9
Software	9
Hardware	10
Capacity	10
Dependencies / External Services	10
Solution Structure	10
Data Access	10
Data Management	10
Security	10
Authentication	11
Authorization	11
Caching	11
Logging	11
Testing	11
Privacy	11
Database Design	12
System Files	12

1. REVISION HISTORY

Date	Version	Author	Description
June 12, 2019	1.0	Gaurav	

2. QUERIES

Type of code used-

- **Backend**
 - Server Side Programming: PHP 5.3+
 - Database: MySQL
 - Framework: Codeigniter
 - Scripting: Java Script, JQuery, AJAX
 - Designing: HTML5 / CSS3
 - Mobile Responsive: Bootstrap
 - Server OS: Cent OS (Linux)
 - STACKS: LAMP

 - **Android**
 - Framework: - Android Studio
 - Language: - JAVA
 - Database: - SQLite
 - Integrated Development Environment: Eclipse/Android Studio
 - Web service: - JSON preferred

 - **IOS**
 - Framework: - Xcode 10.1
 - Language: - Swift
 - Database: - SQLite/Core data
 - Integrated Development Environment: - Xcode
 - Development platform: - MAC
 - Coding Language: Swift
 - Web service:-JSON preferred
-

Unit tests reports -

Ans-Yes test would be provided once we go live

Security checks -

Ans-Will happen on live server

Penetration test -

Ans-Will happen on live server

Technical documentation -

Ans-Will be provided

System architecture documentation -

Ans- Confidential - To be provided on approvals

Documentation on the database and on the organization of the data Functional check of the Alpha Test app Beta private test -

Ans-Confidential-To be provided on approvals

Test of response speed and download speed -

Ans-Will happen on live server

Space occupied on the servers provided both for the part of basic functions and for the archives -

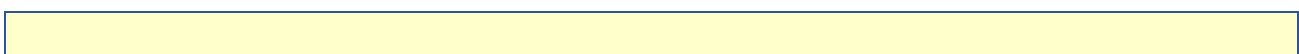
Server Software specification -

Ans-

Disk Space-100GB

IOS-7Mb

Androin-4Mb



3. ABBREVIATION LIST

What all are the abbreviations are used in this document, listed below.

Abbreviation	Description
API	Application Programming Interface
CMS	Content Management System
MS	Management System

4. ROLES AND RESPONSIBILITIES

System design can cross many different groups within an organization to ensure requirements are gathered and met for all stakeholders. As such, a roles and responsibilities section may be necessary to provide the team with clarification on who performs various roles. This section also serves as a list of points of contact for the team and stakeholders should issues and concerns arise which need to be addressed.

5. PROJECT REFERENCES

About this project idea and reference client has provided core document. Based on the document the project we are developing & document link is:

<https://drive.google.com/drive/folders/1oTcvwmpUeiY8-rxFG05E8q-UnlcRu3b>

6. DESIGN CONSIDERATIONS

6.1. ASSUMPTIONS

We assume that customers or users will use updated browsers and with high speed internet because we are using updated softwares like frontend, backend technologies.

6.2. CONSTRAINTS

The website will support the updated browsers and definitely need high speed internet. It support for all devices like phone, tab, laptops & desktops.

6.3. CONTINGENCY

We planning to upload our code in git repository in every weekly.

6.4. STRATEGIES

- Describe any design decisions or strategies that affect the overall organization of the system.*

- Describe the reasoning for each decision/strategy, and how priorities were balanced or traded-off.
- Describe significant alternatives that were considered and the reasons for rejecting them

Design decisions may concern (but are not limited to) things like:

- Use of a particular tool or library
- Future plans for extending or enhancing the system
- Error detection and recovery
- Persistence
- Concurrency
- Communication mechanisms
- Resource management

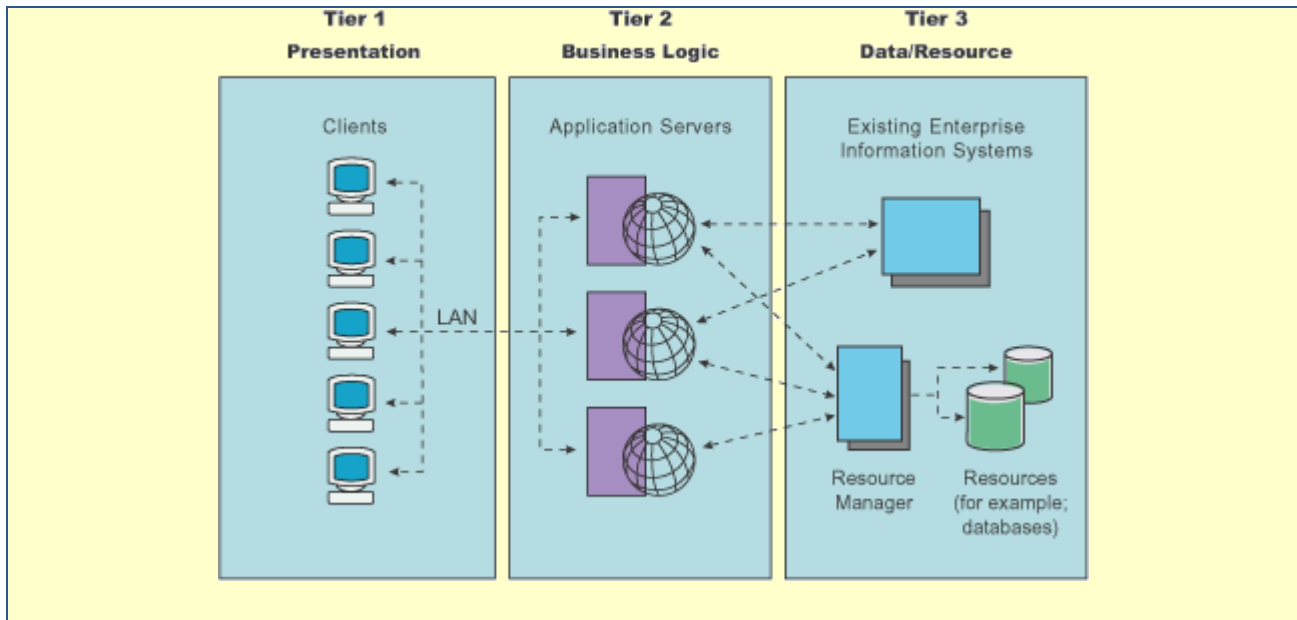
7. TECHNICAL ARCHITECTURE

7.1. SYSTEM OVERVIEW

This website mainly with three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and a well-established software architecture.

As the architecture is three tires, exception handling was done on each layer separately. The layer that was faced with an exception, raise a new exception to the upper layer if needed. The final layer which is UI decides the proper message to show to users and with the current implementation, UI usually redirect the user to a special page that contains the proper message.

Example diagram:



7.2. LOGICAL VIEW

In logical the customers book their trip vehicle with luxuries feasibilities and also shopping feasibilities while traveling with payment and pay later option, it will be including in the final bill.

7.3. COMPONENT VIEW

Provide a high-level diagram/description of the application architecture.

7.4. COMPONENT DESCRIPTIONS

Describe the components that make up the diagram

7.5. TECHNICAL REQUIREMENTS

7.5.1. Software

List all the dependent software used by the application. For example:

- *MySQL Database (4.4)*
- *Atom IDE, Android Studio,*
- *jQuery*
- *Codeigniter*
- *jQuery Validation*

7.5.2. Hardware

For this project we are working with 8 PC's, 4 Android devices & 2 iPhones.

7.5.3. Capacity

For this project need minimum configuration with 500GB hard disk, 8GB RAM & Core i3 Processor.

7.5.4. Dependencies / External Services

If applicable, describe any dependencies on and interoperability requirements for external services

7.6. SOLUTION STRUCTURE

Describe the organization and structure of the Visual Studio solution including the various projects/layers used

- *Domain*
- *Services*
- *Persistence*
- *Reports*

7.7. DATA ACCESS

Codeigniter not provides direct access, so we have use ORM model form and codeigniter is mvc patterns.

7.7.1. Data Management

Describe the following as applicable:

- *data conversion*
- *migration*
- *archival implementation/strategy*

7.8. SECURITY

For security purpose, we are using session, once user logged into site then they can access the data otherwise they have access some guest data only.

Include all security considerations and decision points. For example:

- *Authentication*
- *Authorization*
- *Encryption*

7.8.1. Authentication

For now we are not facing any security issues because we are using MVC structure, it's helps to secure the data and any API call can't access directly. Using model class we can access, for controller can access authenticated users only. For better use remove the keep logged in option.

7.8.2. Authorization

For Admin Panel: We are using session authentication for admin panel. Once the super user or sub users enter the credentials in login time verifying with encrypted data from database. If user credentials matches then we proceeding to session. In session creating random encrypted unique key assigned to session variable, a user can access until the session destroy. Here authenticated user only can access.

For Website: Here also we are using same process, but difference is guest also access some basic guest modules.

For App: In app we are using Rest API, here user can access like some guest module. And other authentication AUTH_KEY & access_token. The AUTH_KEY is a authentication key for accessing every API & access_token is a encrypted unique random generated key for every single users. This is generating in the time of login & the lifetime for 30 days (expiry date) and using refresh token api they can extend expiry date.

7.9. CACHING

We are using both from javascript & php cookies.

7.10. LOGGING

For now admin is taking username and password and comparing with existing but password is storing in encrypted format, from the user password encrypting again.

7.11. TESTING

We are testing while developing a application ie. Unit testing and once development done QA team will test and report.

- Unit Tests
- Manual Tests

8. PRIVACY

Data will be secure outside of the application can't access the data. Almost the other personal data admin can access in this website, other guest can submit their own data and also they can see their data or information.

9. DATABASE DESIGN

The Database Design for the Application is composed of definitions for database objects derived by mapping entities to tables, unique identifiers to unique keys, attributes to columns and relationships to foreign keys. During design and development phase of database, these definitions may be enhanced to in order to support the requirements of the application.

9.1. SYSTEM FILES

*All supporting files are in **"/assets/"** and uploaded file are in **"/assets/uploads/"** for uploading files we are renaming as unique in that directory.*

- *We are now support to upload maximum 4MB.*
- *For file upload we are generating unique name and no need to override.*
- *Backup and recovery from the server will get all files.*

10. SYSTEM INTEGRITY CONTROLS

Finally integration need all the permissions from

- *Payment Gateway verification*
- *Google API Key enable*
- *If SMS gateway needed*
- *Server details & others*

Presented By

PROVAB TECHNOSOFT PVT. LTD.
#236/92/1A, 1st Floor,
Venkataadri IT Park, Electronic City,
Bangalore, Karnataka 560100, India
GSTIN: 29AAF3942G1Z9
