

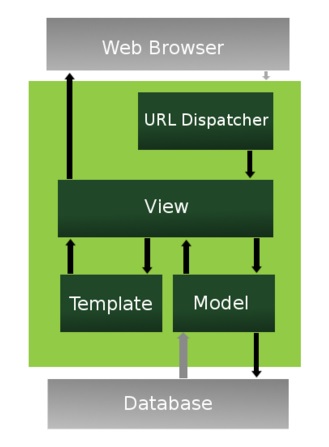
System Design document

Team Wellington



# Architecture

## Web Framework



# Data Access Design

## Model Design (Data Model)

*‘User’ will be a model. The user will have:*

User ID <PK>

*Username*

*Email*

*First name*

*Last name*

*Phone Number*

*Each user may post multiple “listings”*

*‘Listing’ will be a model, each listing will have:*

*Listing ID <PK>*

*Title*

*Condition*

*Description*

*Asking Price*

*Date*

*Time*

*User ID <FK>*

*Each listing may have multiple comments on it, and users may post one or more comments.*

*‘Comment’ will be a model. Each comment will have:*

*Comment ID <PK>*

*Author*

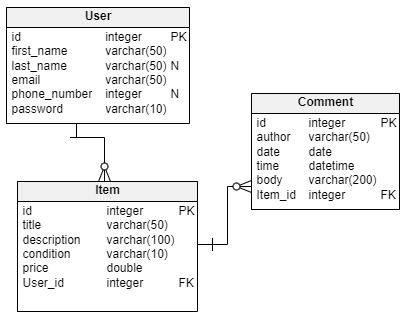
Date

Time

*Comment\_body*

*Item ID <FK>*

*Each user may post multiple “listings”*



## Functional Decomposition

Unregistered User

* Sign Up page to enable registering an account.
* List View page displaying all current listings.
* Detail View page displaying a selected listings detail.
* Contact page displaying companies contact details and a contact form.
* About page showing the companies story and vision.
* Privacy policy page detailing the company's privacy policy.

Registered User

* Will utilize all above functions.
* Login page to allow a registered user to log into their account.
* Logout page to log out of the application.
* Create View page to add a listing.
* Custom Create View to add comments to listings.
* Edit View page to enable editing the logged in users own listings.
* Edit View page to enable editing the logged in user's personal details.
* Delete View Page to allow user to delete their own listing once sold.

Admin Team

* Will utilize all above functions with full authorization.

## Database Investigation

**SQLite**

<https://www.sqlite.org/index.html>

Positives:

* Small
* Fast
* Self-contained
* High reliability
* Full featured

Negatives:

SQLite is fine for small applications or mobile applications but not so good for larger projects. Projects will slow down significantly in big projects.

**Amazon Web Services (AWS)**

<https://aws.amazon.com/?nc2=h_lg>

Positives:

* User friendly
* Scalable
* Reliable
* High performance
* Secure

Negatives:

* Can be expensive and costly
* Prices vary
* There are some limitations
* Complex infrastructure
* A lack of experts

**PostgreSQL?**

<https://www.postgresql.org/about/>

Positives:

* Free and open source
* Powerful
* Reliable
* Feature robustness
* Has reliable performance

Negatives:

The performance of PostgreSQL is slower than MySQL

**Which is more appropriate for this web app?**

For this project, SQLite is not appropriate for this project as large data would affect the performance of the web application. It might slow down the web app and our reputation would be highly affected by the performance.

AWS might be considered appropriate but could be an expensive option. Mostly appropriate for projects with deep pockets and a more complex project than the one we are building.

PostgresSQL is the most appropriate for this web application because PostgresSQL is free and open source. For young startup companies that are just beginning to test the market, it would be best to keep the costs down. Also, PostgresSQL is extremely popular and appropriate for scalable projects that can handle a lot of data.

# Security Design

## Framework Security

Django security features provided:

* Cross site scripting (XSS) protection
* Cross site request forgery (CSRF) protection
* SQL injection protection
* Clickjacking protection
* SSL/HTTPS
* Host header validation
* Referrer policy
* Cross-origin policy (New in Django 4.0)
* Session security
* User-uploaded content

Additional security

* Keeping Secret key, a secret
* Keep python code outside of the web server's root

Django Security features we will use:

* Cross site request forgery (CSRF) protection
* SSL/HTTPS
* Cross site scripting (XSS) protection
* User-uploaded content

## User Roles and Permissions

Unregistered User (site visitor)

* View all active listings
* View About pages
* View Contact page

Registered User (Account holder)

* All permissions available to an unregistered user
* Create a new listing
* Edit users own listings
* Delete own listings
* Comment on any listings
* Edit personal account details

Administration Team

* All permissions above
* Edit any listings
* Delete any listings
* Edit any comments
* Delete any comments
* Edit User accounts
* Delete User accounts

# User Interface Design

## View Design (UI List)

### User Story 1

* “As a new user I need an account sign up page so that I can use all the features of the web application”

### User Story 2

* “As a registered user I need a contact seller page so that I can contact the seller”

## UI Design

Graphical user interface

Description automatically generatedGraphical user interface, application

Description automatically generated