## **High-Level Design**

October 5, 2022

**Issued by:**

Algorithmic Alchemist

**Team Lead**

Sierra Harris

**Team Members**

Bryant Lam

Abhay Solanki

Faisal Al Muharrami

Kevin Thai

David Chan

Github: <https://github.com/abhay772/AA_Senior_Project/>

## Version History

| Version # | Date | Reason for Change |
| --- | --- | --- |
| **Version 1** | 10/5/2022 | Original Document |
|  |  |  |
|  |  |  |
|  |  |  |

## **Table of Contents**

[**High-Level Design**](#_c2950lda9ayi) **1**

[**Version History**](#_yqfturm2nkvt) **2**

[**Table of Contents**](#_28gsa9wbu4q7) **3**

[**Glossary (definitions/abbreviations)**](#_v1knrsbrgvbp) **4**

[**Overview**](#_5gfjcdz7vbk7) **5**

[**High-Level Diagram**](#_k9ovz08fqmbc) **6**

[**Security:**](#_fgq670anoqzs) **6**

[**Input Validation:**](#_te9kychmg7im) **6**

[**Error Handling:**](#_86fbaiyz97aw) **7**

[**References**](#_udqh1rfec7zo) **9**

## 

## **Glossary (definitions/abbreviations)**

* Property Manager: A property owner, commercial property manager, and landlord, anyone that manages the property.
* Service Provider: An organization or company that provides a service.
* PMtoGo: Algorithmic Alchemist proposed single page web application
* DAL: Data Abstraction Layer
* DS: Data Store
* DSL: Data Store Layer
* SL: Service Layer
* PL: Presentation Layer

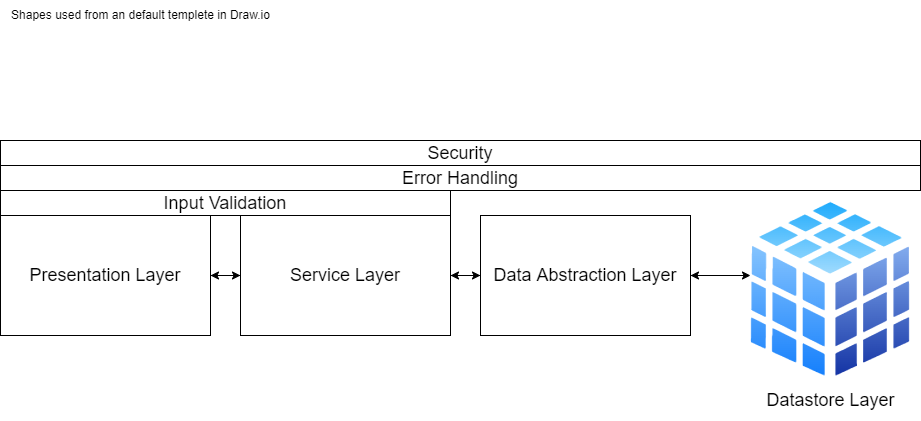
## 

## **Overview**

The high-level design document will describe the general system and architecture of PMtoGo. This document will address the system as a whole and present an visual estimate of the system’s architecture. In addition, this includes how the system handles errors and validates input at a design level.

* Datastore Layer:
  + Datastore layer will consist of one or more types of databases.
  + This will be used to store and access data.
* Data Abstraction Layer:
  + DAL or Data Abstraction Layer will facilitate the extraction of data, from the Datastore, in the form of objects. Instead of data structured as rows, and tables
  + This allows for the abstraction of Business Logic in the service layer.
  + This layer makes adding and implementing new datastores easier.
* Service Layer:
  + Service layer will encapsulate all the business logic for the domain model.
  + This layer handles all data manipulation, accessing, and storing by calling the DAL.
  + And the processing, analysis, and calculation for all the services will be passed on to the Presentation Layer.
* Presentation Layer:
  + This project will be presented as a Single Page Web app.
  + All the UI and UX components will be encapsulated by this layer.
  + This layer will be mainly client facing.

## **High-Level** Diagram

* 

## Security:

* A website firewall will be used to patch and secure any invulnrabilities the website has.
* Users will be advised to enter strong passwords and that will be a validated.
* The website will be safely isolated on a single server.
* Users will be limited according to the business rules.
* The site will have a daily back up to minimize damage incase anything happens.
* The data between host and client will be encrypted during transit.
* During the testing phase the website will undergo security tests to ensure no exploits or vulnerabilities are present.

## Input Validation:

### 

* If an invalid data type is entered or passed, raise “invalid data type” exception.
* Display “Valid characters: A-Z, a-z” for string and character input types.
* Display “Valid characters: 0-9” for integer and float input types.
* Display “Valid characters: A-Z, a-z, 0-9” where only alphanumerics are allowed.
* Display “out of range, {range}” if the input is out of a given range.
* Raise “Valid characters: A-Z, a-z” for string and character input only.
* Raise “Valid characters: 0-9” for integer and float input only.
* Raise “out of range, {range}” if the input is out of a given range.
* Raise “Valid characters: A-Z, a-z, 0-9” where only alphanumerics are allowed.

## Error Handling:

### Database Layer

* + Failure to add data because the parameters not satisfying the primary key will raise, the “Pass all the primary key values” exception.
  + Failure to add data because of too many parameters will raise, “Too many parameters passed” exception.
  + Failure to add data because of invalid data type will raise ,“Invalid data type” exception.
  + Failure to delete data point(s) because of ‘data not found’, will raise the “Data point(s) not found” exception.
  + Failure to modify data point(s) because of ‘data not found’, will raise the “Data point(s) not found” exception.
  + While querying the datastore, if more rows are returned than expected then the query will be terminated and the process/user will raise the “More rows than expected were returned” exception.
* Data Abstraction Layer:
  + If an “Pass all the primary key values” exception is raised, raise, “Required fields are empty” exception.
  + If an “Too many parameters passed” exception is raised, raise, “Too many parameters passed” exception.
  + Failure to add data because of invalid data type will raise ,“Invalid data type” exception.
  + If an “Data point(s) not found” exception is raised, raise, “{object\_name} not found” exception.
  + If an “Invalid parameter passed” exception is raised, raise “Invalid parameter passed” exception.
  + If “More rows than expected were returned” exception is raised, raise “More results than expected were returned” exception.
* Service Layer:
  + If an “Required fields are empty” exception is raised, raise, “Required fields are empty”.
  + If an “Too many parameters passed” exception is raised, raise, “Too many parameters passed” exception.
  + If an “{object\_name} not found” exception is raised, raise, “{object\_name} not found” exception.
  + Failure to add data because of an invalid parameter passing will raise, “Invalid parameter passed” exception.
  + If “More rows than expected were returned” exception is raised, raise “Found more results than expected” exception.
  + If a method/function is not completed, return None object.
  + If an object fails to register with the DAL, raise “{object\_name} not added, try again” exception.
* Presentation Layer:
  + If an “Required fields are empty” exception is being raised, wait till all the required fields are filled.
  + If an None object is returned, display “N/A”.
  + “{object\_name} not added, try again” exception is raised, display “{object\_name} not added, try again”. And load the “Add {object\_name}” form again.

## 

## References

* “Application-Manager-Driver Architecture.” *F´*, https://nasa.github.io/fprime/v1.5/UsersGuide/best/app-man-drv.html#:~:text=The%20manager%20layer%20manages%20a,at%20all%20by%20the%20Application.
* “Data Validation.” *Wikipedia*, Wikimedia Foundation, 22 July 2022, https://en.wikipedia.org/wiki/Data\_validation#Data-type\_check.
* “Exception Handling.” *Wikipedia*, Wikimedia Foundation, 17 July 2022, https://en.wikipedia.org/wiki/Exception\_handling#Exception\_handling\_based\_on\_design\_by\_contract.
* Lteif, Georges. “High-Level Solution Design Documents: What Is It and When Do You Need One.” *Operational Excellence in Software Development*, 8 Sept. 2022, https://softwaredominos.com/home/software-design-development-articles/high-level-solution-design-documents-what-is-it-and-when-do-you-need-one/.
* Publications, Manning. “The Layers of a Cloud Data Platform.” *Manning*, 9 Oct. 2020, https://freecontent.manning.com/the-layers-of-a-cloud-data-platform/.
* Richards, Mark. “Software Architecture Patterns.” *O'Reilly Online Learning*, O'Reilly Media, Inc., <https://www.oreilly.com/library/view/software-architecture-patterns/9781491971437/ch01.html>.
* “Service Layer.” P Of EAA: Service Layer, https://martinfowler.com/eaaCatalog/serviceLayer.html.
* Smith, Jon, et al. “Using Entity Framework with an Existing Database: User Interface.” *Simple Talk*, 17 May 2021, https://www.red-gate.com/simple-talk/development/dotnet-development/using-entity-framework-with-an-existing-database-user-interface/.
* *Website security guide: Secure & protect your website*. Sucuri. (2022, July 27). Retrieved October 5, 2022, from https://sucuri.net/guides/website-security/
* https://ux.stackexchange.com/questions/57491/how-to-clearly-state-to-the-user-what-characters-are-valid