**Webapp Master Document**

| This is a source of truth document. It contains project details, scope, constraints and links to related technical documents and assets of this project. If something is not here or linked from here, it does not exist. |
| --- |

**Index**

[Intro: What is it?](#_hj6k7hiwb5o4)

[Background: What problems does this solve?](#_gv89p9fzk6xq)

[Users: Who will use it?](#_yb8cs2dkxrkc)

[Functional Requirements: What can users do with it?](#_tkc5en8cdpqd)

[Non-Functional & Other Requirements](#_wqthxygb2qne)

[System-generated emails](#_6gcbx1nblxwp)

[Out of Scope: What are we not going to do?](#_4mljrllrsyf7)

[UI Design](#_s901t7u1gj9l)

[System Design](#_i2yedfh1axth)

[Tech Stack](#_pagi7p4pqs17)

[Integration with the Registry & Data Migration](#_gz61m2sma5e7)

[Ideas, features & functionalities for v2](#_vk1220azrtrz)

### 

### 

### Intro: What is it?

An online web application where DPG applicants can submit their applications and the DPGA team can review & manage these applications.

### Background: What problems does this solve?

There are numerous challenges, bottlenecks and opportunities for improvements across the entire application review pipeline.

Here are some of the specific problems the webapp will solve:

1. Reviewers currently have to execute tasks such as maintaining json formatting, duplicating PRs to their local machine, running CI tests, managing PRs on GitHub, etc. While they are necessary, they add no real value to the actual review of the application itself. The webapp will **eliminate all such tasks and free up reviewers’ time & mindspace** so that they can focus on their core job: reviewing the content of an application against the standard.
2. Educational resources about the DPG standard, the submission form, our review process & policies, etc are spread across various pages on the website & GitHub. This results in a lack of clarity about what is required of the applicant, which leads to low-quality & incomplete applications. This forces the reviewers to go back and forth with them to gather missing information. The webapp will help the applicant discover the right information in the right context at the right time. This will **significantly increase the quality & completeness of the application** **leading to faster processing of applications.**
3. Our submission form is lengthy. There are UX issues with it, such as hidden fields, multi-dependent fields, lack of save functionality to return to it later, hard to navigate, lack of correlation between form sections & indicators, etc. One DPG owner even said that they fear having to go through the form again for their refresher application. The form will be restructured and incorporated as part of the webapp. This will **significantly enhance the application submission experience**.
4. We currently use GitHub Projects and PRs as a tool to manage the application pipeline which is less than optimal because GitHub is meant to be used as a code management & versioning tool. Multiple attempts to streamline the review processes have not been very successful. Implementation comes down to reviewers following the SOPs & policies. The webapp will be custom designed keeping the review process and policies in mind. The implementation will **become system-driven rather than human-driven**. This will **improve the efficiency of the review team** as well as increase application throughput.
5. The current application management set up lacks good auditing & status tracking abilities. It’s hard to understand the exact status of an application without consulting the reviewer to understand the missing context. This forces managers to increase their involvement in the review process. The webapp will eliminate this dependency by **empowering managers with monitoring and tracking features by delivering clear status updates in near real-time**.
6. Reviewers currently communicate with applicants manually over email. These email threads often grow into dozens of unstructured emails spread over months. This takes valuable reviewer time to manually monitor the inbox, manage & write the emails. The webapp will **automate & streamline email communications** so that reviewers will not have to even open their inbox.
7. The repos that contain application content do not contain PII data of the applicant. This personal identifying information is maintained separately in a Google Sheet. This separation often creates confusion & staleness of information has previously led to miscommunications with some applicants. The webapp will serve us as a **lightweight contacts book** containing applicant information (name, email, etc) and associate that information with their application, history, etc.

The webapp will do all the heavy-lifting for us, freeing everyone to focus purely on their core competency. It will make our review process scalable and bring everyone on the same page.

### **Users: W**ho will use it?

We are building for 4 types of users

* **Applicants:**
  + They are authorised representatives of digital projects who wish to get their project listed as a DPG.
* **DPGA Reviewers**
  + These are members of the DPGA review team.
  + They review applications and make pass/fail decisions for each indicator.
* **Public Reviewers**:
  + This could be anyone on the internet who is interested in contributing to the review of applications.
* **Admin**:
  + These are members of the DPGA management team who manage the review process and ensure timely review of applications.
  + There will only be one admin account and access credentials will be shared among the relevant members.

### **Functional Requirements: What can** users **do with it?**

* The complete list of functionalities for each user type is maintained on Notion, [please visit this link ↗️](https://shubrank.notion.site/86ec14acfd724fd1a9638a6853a68678?v=5e48750a1b4c4685af503511ea843225)
* The list of functionalities identified are based on the review processes and SOPs outlined here: [Application review process, SOP & policy 2.0](https://docs.google.com/document/d/1vb3DBYgSR8M5DcGhXTmc3ZGUNgT8f4yVL0MMvOCo-tU/edit#)

### Non-Functional & Other Requirements

1. Webapp should be open source. Code should be maintained publicly on GitHub under the DPGA account. OS licence to be used: <to be decided>
2. Webapp should be able to handle upto one thousand simultaneous users to begin with. This traffic includes DPGA reviewers, public reviewers and applicants.
3. Webapp should be hosted on **app.digitalpublicgoods.net**
4. A copy of all applications (except for PII data), including new applications and rejected applications, should be made publicly available in a GitHub repo.
5. A copy of all applications (except for PII data) will be available at a direct public link. Example: app.digitalpublicgoods.net/application/<id>
6. Application IDs will be starting from 10001 and will increase in steps of 1 with each new application.
7. DPG registry should get updated in near real-time based on decisions made through the app.
8. All automated system-generated emails should be sent from an unmonitored email ID.
9. The application form should be publicly accessible without the need for signing up. Signing up should be required to save progress. Email verification should be required to submit an application.
10. Applications in any list should be sorted from largest to smallest score based on the following logic:  
      
    Priority application = 6 points  
    Refresher application = 3 points  
    Clarification submitted = 2 points  
      
    Sorting for applications with the same score will be on the basis who submitted the application first.
11. All activity on the webapp should be time-stamped.

### System-generated emails

This section covers the app’s ability to send automated system-generated emails to applicants.

**List of system-generated emails & email content:** [System-generated email templates](https://docs.google.com/document/d/1jZ4rgrZeBpHyFoI94lYyjM7ZB28Cjl7nMHUFl4-tWKE/edit)

### Out of Scope: What are we not going to do?

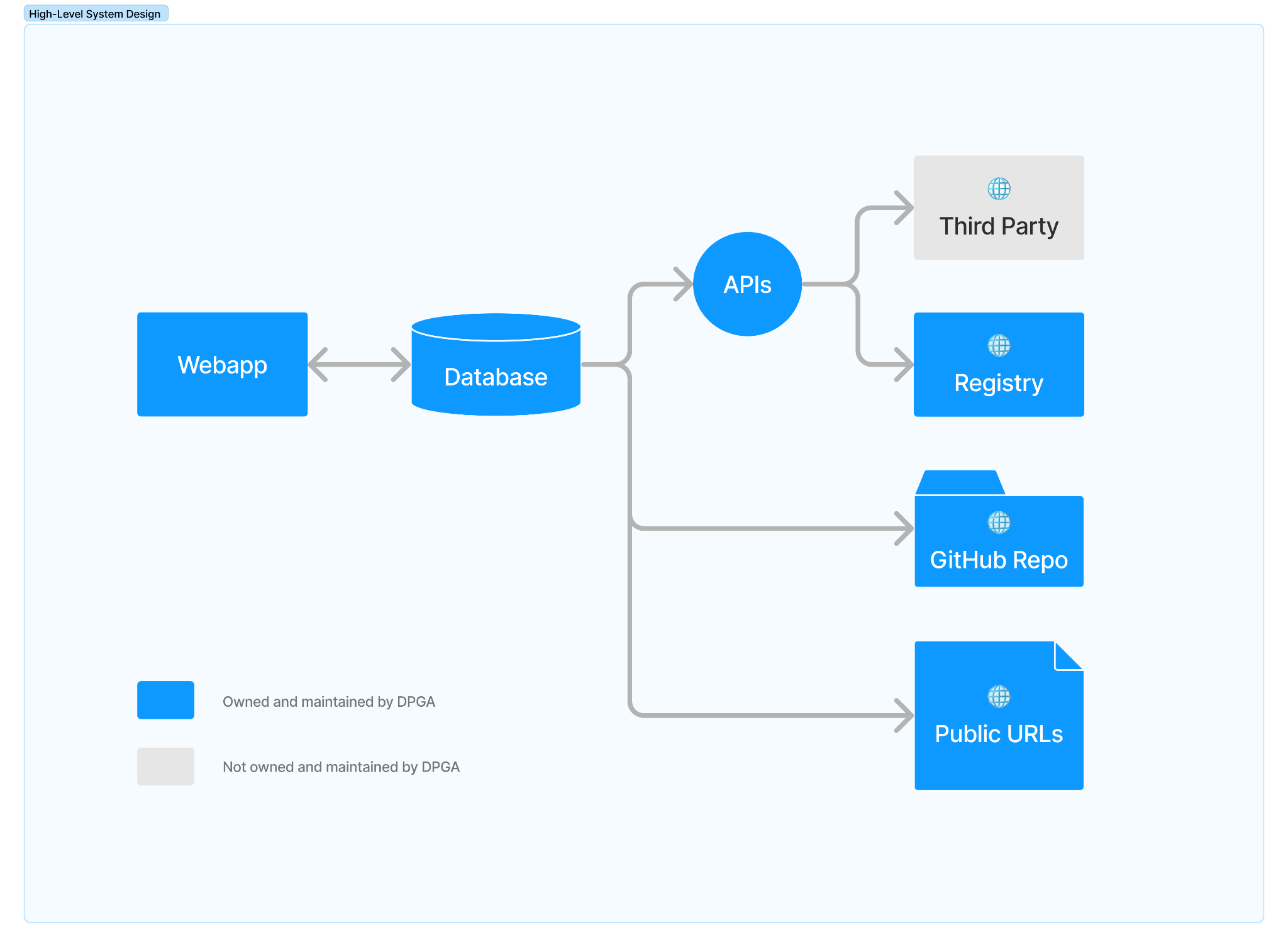
* We will not do any functionalities for experts. Communication with them can take place off-webapp on other platforms such as Slack, Zoom, GitHub comments, etc. Consultants will still be able to access any applications content via public links or GitHub repo.
* We will not focus on building a pretty UI. Focus will be on utility and performance of the app. We will utilise out of the box UI components of a library.
* We will not build a chat or chat-like functionality between any user. This specifically includes ability for reviewers and applicants to communicate back & forth over the app.
* We will not optimise all pages to be mobile responsive, except for pages that are applicant-facing. This is because the majority of work done by reviewers & admins happens on their laptops.
* We will not make multiple admin accounts. Only one admin account should be provided that can be accessed by whoever requires it.

### UI Design

[Click here to view sitemap (page hierarchy) in FigJam](https://www.figma.com/file/7i8UXRgTYtJT3hgQPvoauj/DPGA-Webapp-%3E-Sitemap?node-id=0%3A1)

[Click here to view hi-fidelity UI design in Figma](https://www.figma.com/file/0bVAXlbzBfQloie0OIKM2c/DPGA-WebApp-%3E-UI-Designs?node-id=0%3A1) (more pages will be added to this file as they are designed)

### System Design



### Tech Stack

* Backend - Codeigniter (PHP)
* Database - MySQL
* Frontend - Bootstrap 5
* APIs - REST APIs
* Hosting - Apache server with MySQL (basic hosting or AWS EC2 server)

### Integration with the Registry & Data Migration

Successfully implementation to the webapp depends on migrating the old data and making the necessary changes to the existing properties that will be affected by this switch.

Following steps are proposed:

1. **Map the old API schema to the new database schema.**
   * Mapping available here: [API Schema Mapping for Applicant Data Migration](https://docs.google.com/spreadsheets/d/1k6RSCxAFpylsPblTFbAIiJTPuEZUHLeiEzrdfXNZx_Q)
   * Sources
     + all DPGs: <https://api.digitalpublicgoods.net/dpgs/>
     + all Nominees: <https://api.digitalpublicgoods.net/nominees/>
     + all other solutions: Any solution that exists in the [candidates repo](https://github.com/DPGAlliance/publicgoods-candidates) but is not covered in the above two sources, including applications that are new, under review, etc.
   * This step will be executed by [Shubrank Mukhiya](mailto:shubrank.mukhiya@gmail.com)
2. **Fetch existing data using the existing APIs & push it into the new MySQL database.**
   * This step will be executed by [govind.prajapat.39@gmail.com](mailto:govind.prajapat.39@gmail.com)
3. **Manually check & correct discrepancies for all applications, nominees and DPGs. This will be required because of the fundamental differences between the old and new schemas (number of questions, type of response, dependencies, sections, etc).**
   * This step will be executed by [nbaleeta@gmail.com](mailto:nbaleeta@gmail.com)
4. **Rewrite the APIs**
   * This step will be executed by [govind.prajapat.39@gmail.com](mailto:govind.prajapat.39@gmail.com)
5. **Make necessary changes to the registry to work with the new APIs.**
   * The extent of changes will be limited to refactoring of how the data is pulled (using new API instead of old ones)
   * This step will be executed by @nathanfletcher

**Important considerations**

* What happens to the registry?
  + No major changes except the ones required to enable consumption of new APIs to keep displaying the data on the registry.
* What happens to the existing GH repos at:
  + <https://github.com/DPGAlliance/publicgoods-candidates> -> will be preserved because of historical data like comments, activity log etc it holds. A PR will get created with each new application in this repo but the repo ceases as the place to review the application. It will only be used for public discussion using GH comments, if required.
  + <https://github.com/DPGAlliance/publicgoods-submission> -> could be archived as it will no longer be relevant.
  + <https://github.com/DPGAlliance/publicgoods-scripts> -> some parts of this will no longer be relevant
  + <https://github.com/DPGAlliance/publicgoods-form-opens-pr> -> will have to be either updated or archived depending on how we deal with the candidates repo.
* What happens to existing APIs?
  + APIs will be rewritten to work with the new schema.
  + API location: <https://api.digitalpublicgoods.net/> -> We can continue to host them at the current location
  + API documentation: <https://github.com/DPGAlliance/publicgoods-api> -> Will be updated.
  + Data source: <https://github.com/DPGAlliance/publicgoods-api/tree/main/docs> -> new MySQL database
  + Third party such as DIAL that are utilising existing APIs will have to be contacted to make necessary updates on their side to keep pulling the data via the new APIs
* What happens to the existing public URLs of each DPG?
  + A copy of each submitted application is made available via a public URL so that Experts can quickly access an application in order to collaborate with reviewers and the general public can access an application in a user-friendly format.
  + These webpages are generated by the webapp in this format : app.digitalpublicgoods.net/application/<application-id>
  + All existing public URLs, example: https://digitalpublicgoods.net/registry/<id>.html, will be removed.
* What happens to the google sheet [Contact Information For DPG Submissions](https://docs.google.com/spreadsheets/d/1cWklJYm-thSQGG9JAYjcZtzSf4Tf6TAzf3DYEBUSpBs/edit#gid=0)?
  + The webapp will serve as a lightweight contacts book of all applicants, so contact information in the sheet will be absorbed in the webapp.
  + The sheet will be saved but not updated going forward.
* What happens to old applicants?
  + We will create a user account for every current DPG and Nominee by using the info in this sheet [Contact Information For DPG Submissions](https://docs.google.com/spreadsheets/d/1cWklJYm-thSQGG9JAYjcZtzSf4Tf6TAzf3DYEBUSpBs/edit#gid=0)
  + They will need to be informed of this and instructions to gain access to their account by forgetting the password.

### Ideas, features & functionalities for v2

* Ability to write in rich text in the application form.
* There should be a handy checklist next to each indicator for DPGA reviewers for quick access and recall.
* Granular “late” tags
  + Highlight an application if it has been in **Waiting for L1** for more than **8 days**.
  + Highlight an application if it has been in **Waiting for L2** for more than **8 days**.
  + Highlight an application if it has been **Under Consultation** for more than **15 days**.
  + Highlight an application if it has been more than **30 days** and a decision is still pending.
    - Since the date of submission of application.
    - If clarifications are requested, this timer will stop and restart from the date of submission of clarifications.