# 

# 

COMPSCI 732

Project Group

Proposal

# 

Group members:

Leo Li - 224693733

Shuming Wang - 966680384

Chenghao Liu - 872374103

Tuanyu Zhou - 283572245

Yichao Yan - 695603634

Zhanghang Liu - 255142352

# **Table of Content**

[1. Introduction](#_ltkftxfgo33k)

[2. Related work](#_75i7ljcr7gar)

[3. Requirements](#_irf5w5uqoi0)

[3.1 Must-have](#_r4g2ftckj1eh)

[3.1.1 Web-based Feature Flag management platform](#_jp09gu3spuk0)

[3.1.1.1 Feature Management](#_ib8tijvr9c3t)

[3.1.2 Demo website using this feature flag system](#_kh9vl8ah0i6f)

[3.2 Should-have:](#_q9zajsmzglmb)

[3.3 Could-have:](#_tgfvsy6lxgzm)

[3.4 nice-to-have:](#_nz5xcqexn44v)

[4. Technologies](#_908mi03bfs53)

[5. Project management](#_7r9vcn6mkx6m)

[5.1 Project management strategy:](#_tb6stqxg2lbw)

[5.2 Communication and coordination of our team:](#_1hi1eyvu8pcy)

[5.3 Version control strategy:](#_q48jlqk3on1t)

[6. Risks in team projects](#_wt70wgwm41vm)

[6.1 Risk factors of individuals](#_3mhcqg6asf71)

[6.2 Risks factors between team members](#_lb8kihmc0wb0)

[6.3 Risk factors of the project](#_42ev5bizgyj4)

[6.4 Mitigation measure](#_et46bj3uauqc)

# 

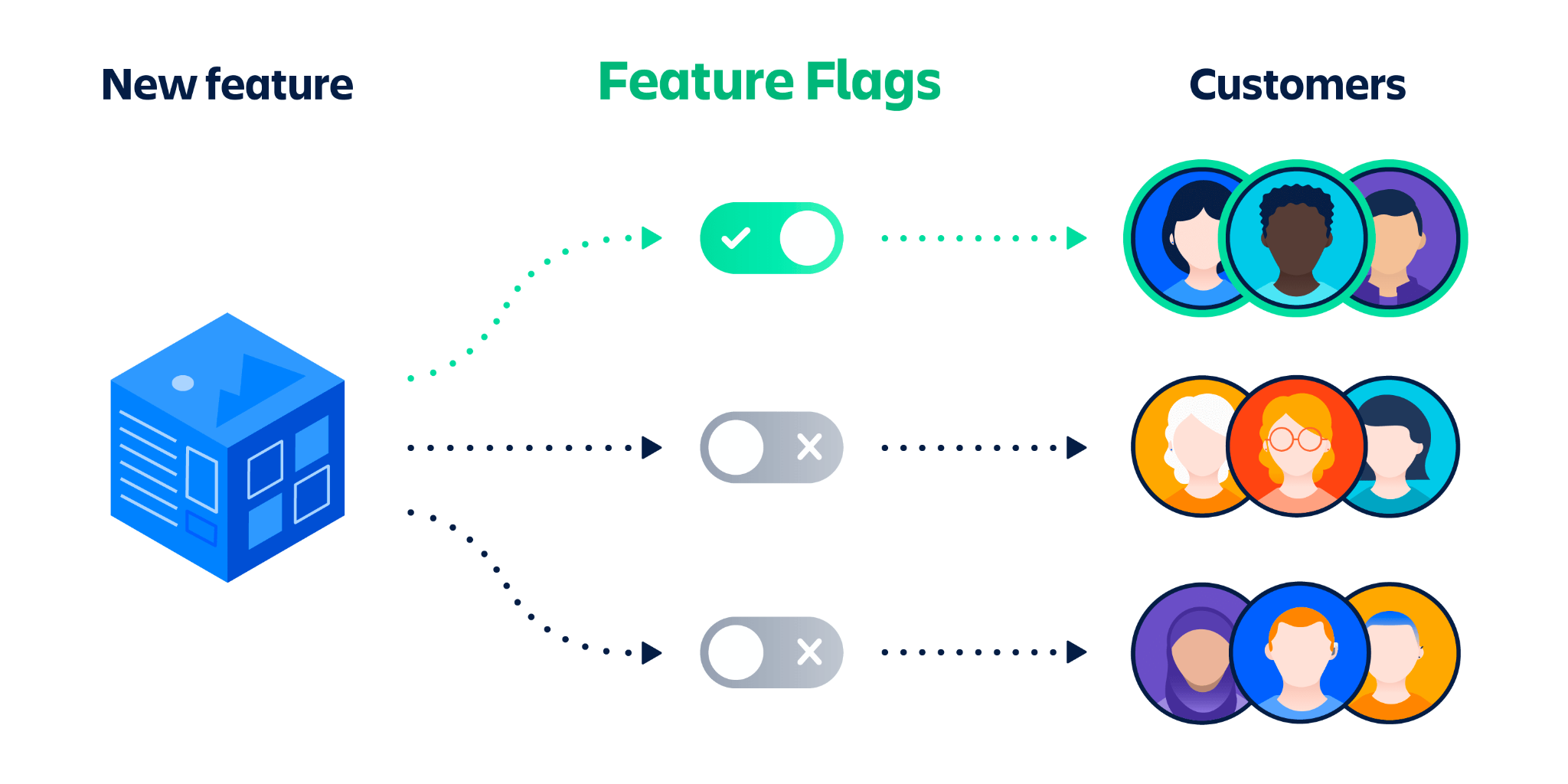
# 

# 1. Introduction

In recent years, the rapid advancement of technology has resulted in the development of various software technologies that cater to the needs of engineers, thereby simplifying their work processes and website management. The project for our team is a feature flag system, which facilitates the management and categorization of data information for users with different permissions. Feature flags are software development techniques used to enable or disable certain features of an application or service without the need for a full deployment or release. Feature flags work by controlling the activation or deactivation of a specific feature through a configuration file or UIs, rather than hardcoding the feature into the application code. The configuration file or UIs contains a list of feature flags, which are associated with specific features of the application. When the application runs, it checks the configuration file or feature flag states in data stores to determine which features are enabled or disabled. If a feature flag is set to "on," the corresponding feature is enabled and can be used by users. If a feature flag is set to "off," the corresponding feature is disabled and cannot be used by users. Our Feature Flag System will be developed to replace the outdated and inefficient work model that required frequent or cumulative releases, as well as the tedious process of accessing all the data during troubleshooting. It is designed to improve data management and enable users to adjust data dynamically, providing a more efficient and user-friendly solution.

**Figure 1**

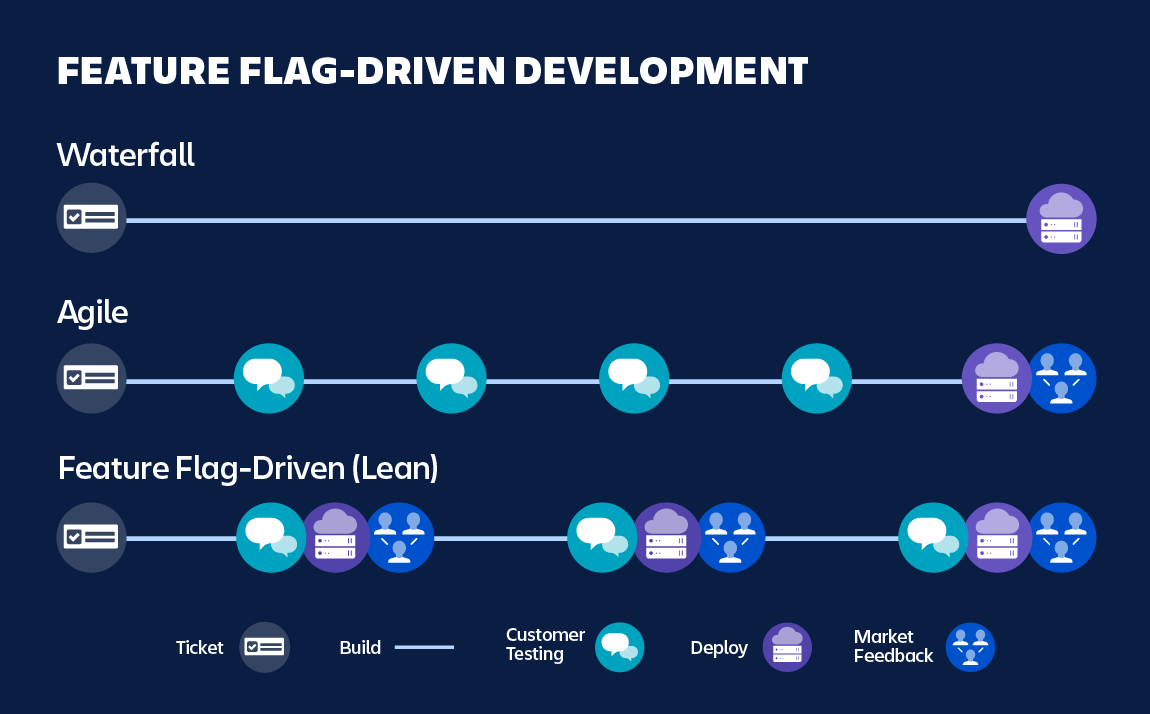
*How Feature flags affect user experiences*

**

*Note*. From by Ian Buchannan, (<https://www.atlassian.com/continuous-delivery/principles/feature-flags>)

**Figure 2**

*Benefits of Feature flag*

**

*Note.* From by Ian Buchannan,

(<https://www.atlassian.com/continuous-delivery/principles/feature-flags>)

# 2. Related work

While LaunchDarkly has been a popular choice for many years, Unleash has emerged as a strong competitor in the market. Like Flagr, Togglz offers similar features and capabilities, but it also has some advantages. Their solutions are very mature, and all have a certain share of the market. LaunchDarkly is a commercial feature flag platform and supports multiple programming languages. Additionally, Unleash is an open-source framework written by Nodejs and the main advantage is extensibility, which can support multiple functional labeling strategies. Flagr and Togglz are written by Go and Java respectively which offer the same service. However, the frameworks mentioned above have some shortcomings in different areas. Some of them are commercial platforms that may bring a burden to startups. In addition, high concurrency may become a problem due to programming language limitations such as Nodejs. Besides, some frameworks lack user-friendly UI and API, which may cause low development efficiency. Therefore, our project aims to fix these shortcomings and support high concurrency. Table 1 shows the difference between different platforms.

**Table 1**

*Comparison of the advantages and disadvantages of different frameworks*

| Feature/Tool | LaunchDarkly | Unleash | Flagr | Togglz |
| --- | --- | --- | --- | --- |
| Open source | No | Yes | Yes | Yes |
| Language support | Multipule | Multipule | Multipule | Java |
| User management | Yes | Yes | Yes | No |
| Scalability | High | High | Medium | Low |

# 3. Requirements

We plan to implement the core features of a feature flag system due to the time limit.

## 3.1 Must-have

## 3.1.1 Web-based Feature Flag management platform

#### 3.1.1.1 Feature Management

* + - New Feature Flag: Create a feature flag to manage who can see a particular product feature.
    - Feature Toggle: Users can enable or disable features at runtime without deploying the whole system. Users should code some predicates (if/else) in their front end.
    - Role-based Toggling: The manager can edit the Feature Flag to specify which group of users can see this feature.
    - List Feature Flags: This page shows all feature flags in a project.
    - Show the detail of a Feature Flag
* User management: Features are mainly controlled by user groups, so we need to develop a page that can manage users, including list users, create user groups, delete user groups, etc. .
* APIs: We will provide APIs for users to call. Users can use these APIs to control the web page’s behaviour.

## 3.1.2 Demo website using this feature flag system

A beautiful website where students can register and browse the page.

* Log on & Login: Because we need to demonstrate our feature flag, we need students registered in a different group, such as the test team, development team, and pre-sale team.
* QR Code: We should embed our URL in the QR Code.
* Simple features that can be turned off through the Feature Flag system.

## 3.2 Should-have:

* Feature Flag Search: In a search box users can search the feature flag by name.
* Strategy-based Toggling: Implement custom predicates (Strategy Pattern) to evaluate if a feature is enabled.

## 3.3 Could-have:

* Feature History
* Powerful test analysis with support for binomial, count, duration, and revenue metrics.
* Features Monitoring: For each feature execution, we evaluate the predicate therefore, it's possible to collect and record events and metrics to compute nice dashboards or draw curves for features usage over time.
* Audit log: The system can track changes of users made in this environment.
* Comparing and copying flag settings between two environments.

## 3.4 nice-to-have:

* SDKs for other languages, such as PHP, Ruby, Python, Go etc. Users can use the feature flag by calling our SDKs.

# 4. Technologies

The project will use the following technologies and materials:

* MongoDB/Atlas - a NoSQL database management system that will be used for data storage.
* Message Queue (no decided): A message queue is a technology that enables asynchronous communication between different parts of an application or between different applications
* Cache: Redis: Redis is an in-memory data store that can be used as a cache, a message broker, and a key-value store.
* Go (Gin framework) - a programming language and web framework that will be used for building the backend of the application.
* Vite/Webpack, React, Redux, redux-thunk - front-end technologies used for building the user interface of the application.
* Github actions - a continuous integration and deployment tool used for automating the build, test, and deployment process of the application.
* Docker - a containerization platform used for packaging and deploying the application.
* Kubernetes (helm) - an open-source container orchestration system that will be used for automating deployment, scaling, and management of containerized applications.
* Terraform - an infrastructure-as-code tool used for managing cloud resources and deploying infrastructure.
* GCP (GKE) - Google Cloud Platform is a cloud computing platform that will be used for hosting and deploying the application.
* API: Restful/ GraphQL - RESTful or GraphQL API will be used for communication between the frontend and the backend of the application.

In addition to the above technologies, we will also research and apply other libraries and frameworks that can help in building a more efficient, scalable, and robust application. Independent learning and research will be emphasized to ensure we use the most appropriate and suitable technologies for the project. Node.js is not used in the project. The reasons why we would choose Go over Node.js for this project are:

* + - Performance and Concurrency: Go was designed to be highly performant and provide robust concurrency support, which is essential for building scalable, high-performance applications. Go uses lightweight threads called goroutines, which enable developers to build highly concurrent applications without the need for complex locking mechanisms. This makes Go well suited for building applications that require high performance and scalability.
    - Lower Resource Requirements: Go is a compiled language, which means that it does not require a runtime environment like Node.js does. This makes Go applications more lightweight and efficient in terms of memory and CPU usage.
    - Large-Scale Application Support: Go has proven to be a reliable choice for building large-scale applications. Its performance and concurrency features make it ideal for building distributed systems, microservices, and other complex applications.

# 5. Project management

## 5.1 Project management strategy:

Our team plans to adopt Agile with 5 sprints, each lasting two weeks. Agile is a popular project management strategy in project development that breaks a large project into smaller manageable tasks that can be delivered within a short period of time. At the same time, Agile requires periodic testing throughout the project lifecycle. This not only allows problems to be identified and corrected by Agile, but also means that the final result will go through many improvements when it is ready for delivery. This approach to project development management is flexible and iterative development, promotes frequent communication between all parties, and improves transparency and visibility throughout the project lifecycle. This makes it easier for our team to adapt to changes as the project progresses.

## 5.2 Communication and coordination of our team:

Our team chooses Slack/Wechat/Teams as messaging apps to communicate with each other. At the same time, we will have online or offline face-to-face meetings according to team members' free time every week, which enables us to effectively coordinate and communicate during the project. These methods of communication can help us ensure that everyone stays in sync and important information is communicated in a timely manner.

## 5.3 Version control strategy:

For the version control strategy, our team chooses to use Git, which is a widely used distributed version control system. Git allows developers to work from a locally copied codebase and then merge their changes back into a central codebase. Git has many advantages, including the ability to work offline, synchronization of code, better collaboration, and easier code review.

# 6. Risks in team projects

## 6.1 Risk factors of individuals

* Too steep learning curve: Our team members come from diverse backgrounds in Information Technology, resulting in varying coding capabilities. Assigning tasks that require significant self-study to gain new knowledge may lead to struggling team members who cannot keep up with the project's pace and complete on schedule. If the technology used in the project is too difficult to master within a short time, it can negatively impact the project's overall quality.
* Loss of focus: each team member has also enrolled in other courses and needs to dedicate significant time and energy to studying. When there are too many tasks to complete, individuals may struggle to focus on other responsibilities, leading to distractions or delays. These distractions and delays can eventually result in low-quality outcomes.

## 6.2 Risks factors between team members

* Poor communication: although our team members can talk online to each other at any moment and have a schedule of regular meetings, poor communication can cause misunderstandings, confusion, inefficiency, and even conflicts within the team.
* Unequal distribution of workload: some experienced team members, who can complete their work rapidly and efficiently, may be over-reliance and take on much more work than others, which can lead to unfair situations, as well as risk concentration. If problems arise with the task completion of these core members, the overall quality of the team project would be greatly affected.

## 6.3 Risk factors of the project

* Cannot be completed on time: due to some reasons like invalid planning, inaccurate estimation, skill constraints, or any other unforeseen events, the team may fail to take corrective action to ensure successful project completion on time.
* Uncontrolled project scope: scope creep can happen since this is a project in which we can choose our own topic and have a lot of space for free operants. Any change in idea or object may lead to an uncontrolled expansion of a project's original goals.

## 6.4 Mitigation measure

Our team will try our best to mitigate these risks. Such as

* To craft and execute a specific, realistic, verifiable plan: to plan in advance about the regular meeting, definition of the project topic, breakdown of the tasks, resource availability as well as a review of the schedule.
* Establish clear project priorities: clearly define the project priorities for all members. This can help ensure that everyone is working on the same level and progressing towards the same goals.
* Keep efficient communication: our team will communicate with each other clearly and frequently to ensure that everyone is aware of the current tasks, progress, issues, and next steps. We will discuss and collaborate to bring technology, innovation, and creativity, especially when some obstacles arise beyond one’s capability.

# Reference

Buchannan, I. (n.d.). *Feature Flags How to Progressively Expose Your Features with FeatureFlags.*<https://www.atlassian.com/continuous-delivery/principles/feature-flags>

*FF4J - Feature Flipping for Java*. (n.d.). Github. <https://github.com/ff4j/ff4j>

*Growth- Open Source Feature Flagging and A/B Testing*. (n.d.). Github.<https://github.com/growthbook/growthbook>

*Your First Feature Flag*. (2023, February 27). Launchdarkly.Doc.<https://docs.launchdarkly.com/home/getting-started/feature-flags>