From Design to Delivery
A Software Engineer's Handbook:
PollEv.com/ghc520

PollEv.com/ghc520

Design Components:

Component Types Overview

Backend:
- **Database**: a permanent, large data store
- **Cache**: A quick-access, smaller data store that's usually a subset of a database. Needs to be refreshed if filled up
- **Load balancer**: A filter-like component that distributes requests to different instances of services so any one doesn't become overloaded
- **Services**: A collection of code that is always accessible, ready to accept requests, perform an action, and return a response
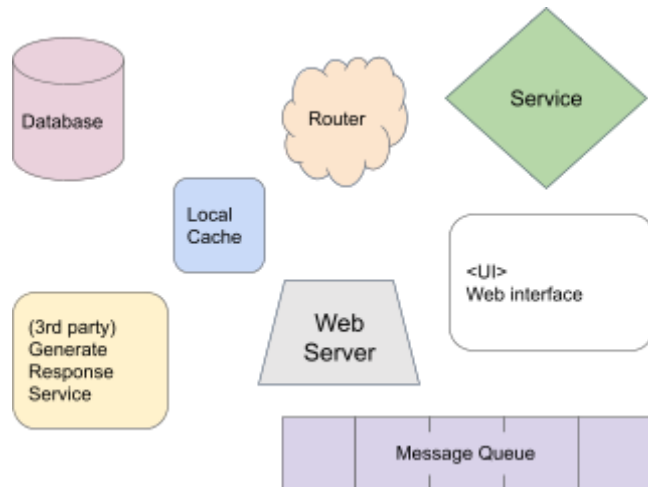
Middleware:
- **Queue**: A collection of messages stored in FIFO (first in first out) pattern, usually used to communicate between services
- **Logging**: A mechanism to track what happened and what's happening in a component in real-time for debugging, visibility, possible crash prevention, etc.
- **Routing (HTTP/GraphQL/etc.)**: Various network protocols to send/receive requests/responses depending on various factors - traffic, data format, etc.

Deployments:
- **Local**: On one machine, usually your development machine
- **Datacenter**: Many machines in one location that your app can run multiple instances on
- **Cloud**: Many machines in many locations that your app can scale even more instances on
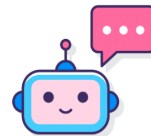
Workbook Activity:

Draft system designs in incremental stages. Start simple!

Database

Router

Service

Local Cache

(3rd party) Generate Response Service

Web Server

<UI> Web interface

Message Queue

Our Client Requirements

Who: Surveyed GHC'22 participants

What: A chatbot to answer questions about GHC'23!

Simple text Q&A automated bot

Chat history up to 1 month

Tens of thousands of clients using this chatbot around the world

Technical Considerations

Different teams and businesses may have different values/considerations:

- **Simplicity** ➜ decoupled, single-responsibility
- **Scalability** ➜ how to grow without impacting the system
- **Reusability** ➜ can we reuse this in the future?
- **Tech stack considerations** ➜ what do we already have?
- **Community** ➜ is it easy to get help or questions answered? Is it a well-documented technology?

Use this space for your system design

Workbook Activity:
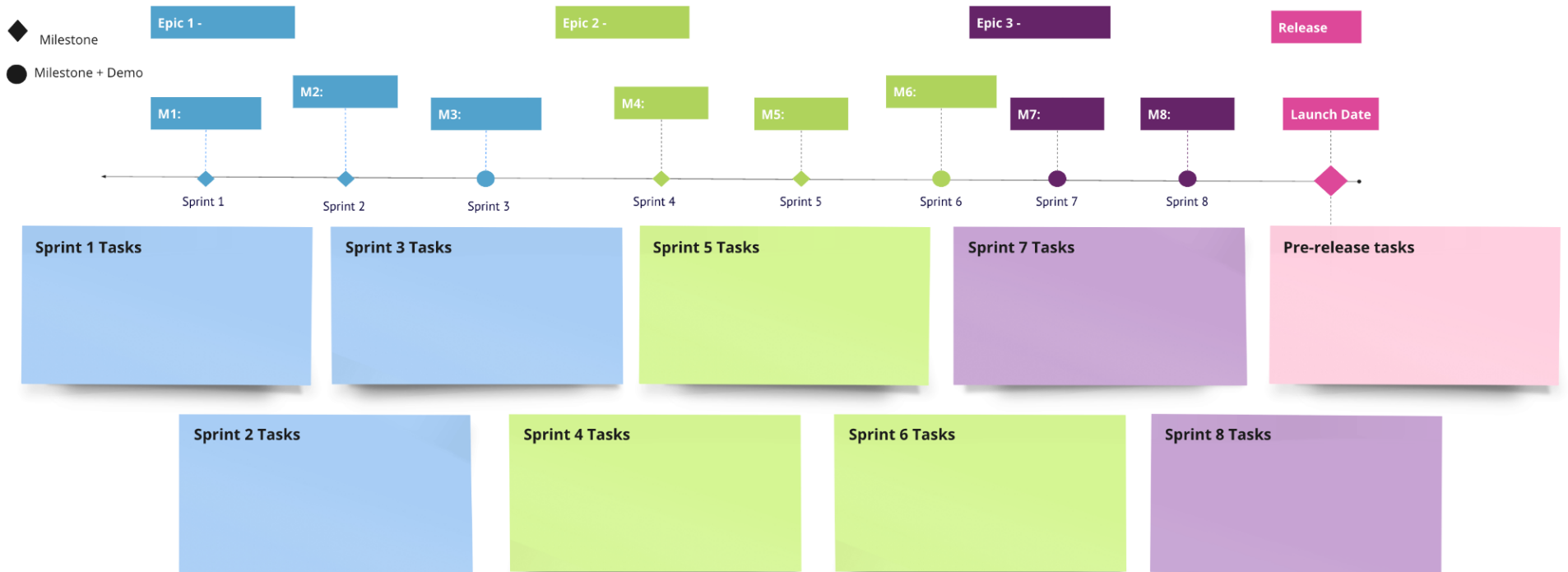
Create Timeline w/ milestones, dates

Assumptions:
Team members: 3 Software Developers
External Support: Product Managers, UX Design Group, Quality Assurance Testers
Deadline: 3 months
Below is a template you may use (or create your own!)

ChatGHC Project Timeline

◆ Milestone

● Milestone + Demo

Epic 1 -

Epic 2 -

Epic 3 -

Release

M1:

M2:

M3:

M4:

M5:

M6:

M7:

M8:

Launch Date

Sprint 1

Sprint 2

Sprint 3

Sprint 4

Sprint 5

Sprint 6

Sprint 7

Sprint 8

Sprint 1 Tasks

Sprint 3 Tasks

Sprint 5 Tasks

Sprint 7 Tasks

Pre-release tasks

Sprint 2 Tasks

Sprint 4 Tasks

Sprint 6 Tasks

Sprint 8 Tasks

Types of Tests:

Types of tests

Unit Tests

- Verifies functionality of a single component in isolation
- Ideally, mocks out external dependencies/side effects
- Useful for Test-driven Development (TDD)

Integration Tests

- Verifies how parts of your system work together
- Tests side effects and connections to external dependencies (i.e., databases, client APIs)
- Catches interface errors between different components

System Tests

- Verifies your software functions correctly as a whole
- Tests entire workflows and client use cases
- Helps identify errors in the system
- Can include non-functional testing (e.g., performance, usability, scalability)
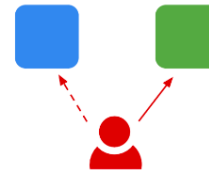
Types of Deployment:

Types of Deployment

Full Deployment

Release all components to everyone at the same time

- Simple, fast, cheap
- Use for low-risk or non-critical releases
- Riskiest: difficult to rollback

Blue-Green Deployment

Deploy to a staging environment and test before switching users to this environment

- Easy to rollback
- Expensive to maintain a replicated production environment

Canary Deployment

Incrementally release application to subsets of users by stages

- Allows testing of real user traffic and side-by-side comparison of different service versions
- Fast and safe to rollback
- Complicated to implement

Workbook Activity:

Match the deployment strategy to the appropriate situation
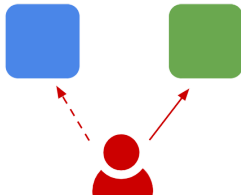
Situation A
The MVP of ChatGHC has to go out to production in 1 week. We don't have any deployment tooling set up yet and will likely only have 10% of expected traffic since our advertising team has not announced an official launch of this App yet.

Situation B
We have 1 month till our release of ChatGHC. This will be a large release with seven microservices, new databases and critical features! We already have over two-thousand clients using our app, so it'd be great to release this seamlessly without causing issues to these clients. Our QA team has also created some test cases to verify the functionality of these new features.

Situation C
In two weeks we want to release the searchable chat history feature. This is a highly requested feature, but we want to be conservative with the release to ensure our big clients don't experience problems/outages. It would be nice to have our smaller clients try out this workflow in production before introducing it to our big clients.

Situation D
The translation service is now ready for deployment! This service will help our spanish users translate the messages from english to spanish, but service should not impact any other functionalities or services.

PollEverywhere:

What metrics to collect?

PollEverywhere:

What's concerning in the logs?

Memory and CPU Usage
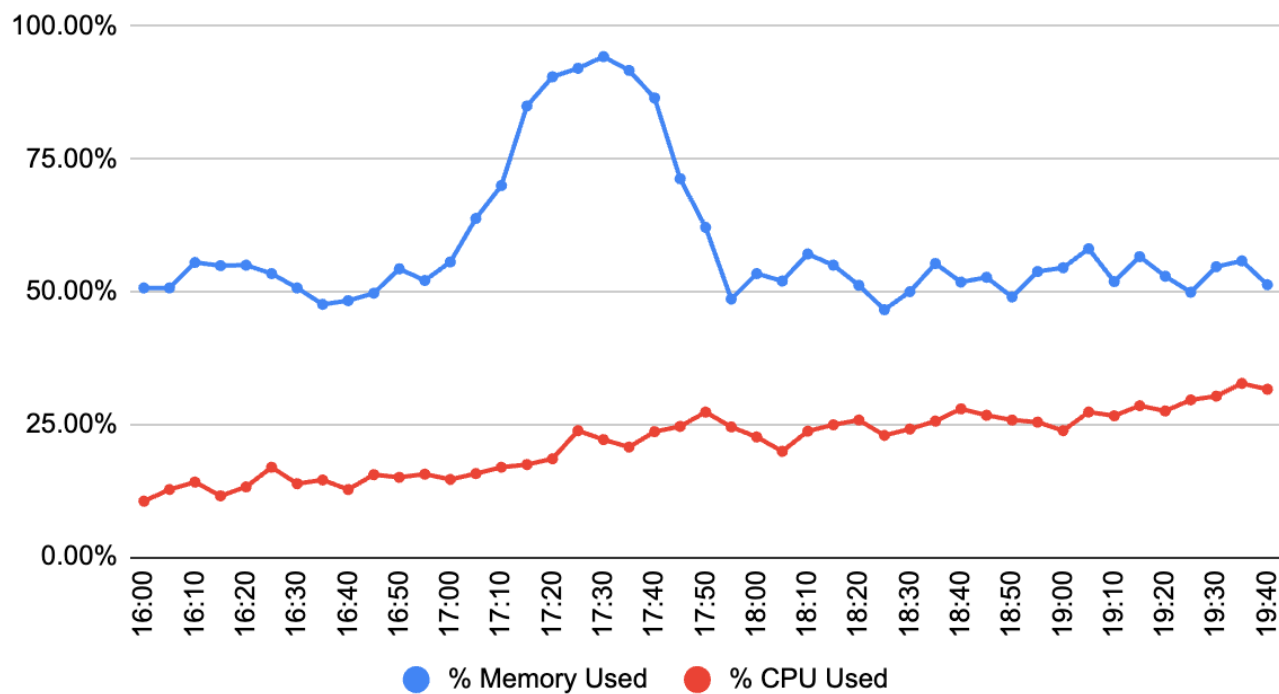
What's concerning in the logs?

	timestamp	level	message
1	2023-09-01 14:00:01.074	INFO	*chatbot_app.py:13* Incoming request from uuid=**439323** request = { sessionID: **a3nf8s3**, question: "How do I get to the convention centre from Rosen Plaza Hotel?" }
2	2023-09-01 14:00:01.083	DEBUG	*chatbot_model.py:27* Generating chatbot response….
3	2023-09-01 14:00:02.192	INFO	*chatbot_model.py:32* Response found for question: "How do I get to the convention centre from Rosen Plaza Hotel?", answer: "To get to the GHC convention centre from Rosen Plaza Hotel please follow these directions: www.tinyurl.com/12$wkds239rfj35. This is a **10** minute walk."
4	2023-09-01 14:00:02.554	INFO	*chatbot_user.py:68* User settings for uuid=**439323** found name="Sarah Smith", defaultLanguage=**e_SPANISH**, isGHCPresenter=**false**, GHCId=**null**
5	2023-09-01 14:00:02.555	WARN	*chatbot_response.py:401* This user has not specified their GHCId
6	2023-09-01 14:00:02.640	INFO	*chatbot_repsonse.py:432* Sending chatbot response to ghcTranslationSvc with request = { text: "To get to the GHC convention centre from Rosen Plaza Hotel please follow these directions: www.tinyurl.com/12$wkds239rfj35. This is a **10** minute walk", language:**e_SPANISH**}
7	2023-09-01 14:00:07.943	INFO	*chatbot_translation.py:112* Unexpected response received from ghcTranslationSvc response = { status:"Error", returnCode: **435**, message: "unable to translate 12$wkds239rfj35. Word does not exist in translation mapping" }
8	2023-09-01 14:00:08.021	DEBUG	*chatbot_response.py:439* Sending chatbot response to user…
9	2023-09-01 14:00:08.113	ERROR	*chatbot_response.py:543* Unexpected exception thrown !!!
10	2023-09-01 14:00:09.120	INFO	Total elapsed time processing request: **7046.23** milliseconds