

KARD

phase 2 presentation

k.

1. An Introduction to kard

Let's start with the first set of slides

k.



Kard

Is a digital business card application that seeks to bring making connections into the 21st century.

k.

Main Features

Create a kard account

Customize your kard profile

Add kards through your kard app



2. A demo of kard

Let's look at the App!

k.



6

3. A demo of kard

Let's look at the CLI!

k.

“

k.

4. Kard Accessibility

Accommodation, simplicity, and contrast

k.

Accommodation

Equitable use

Flexible use

Tolerance for error



Simplicity

Simple and intuitive use
Low physical effort



Contrast

Perceptible information

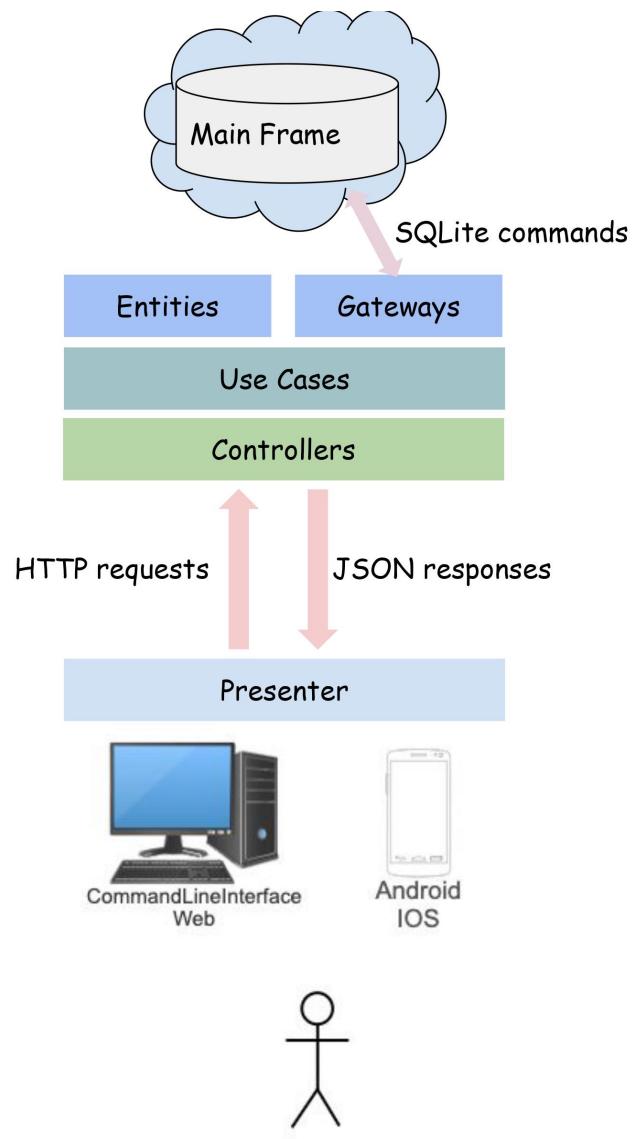
Size and space for approach and use



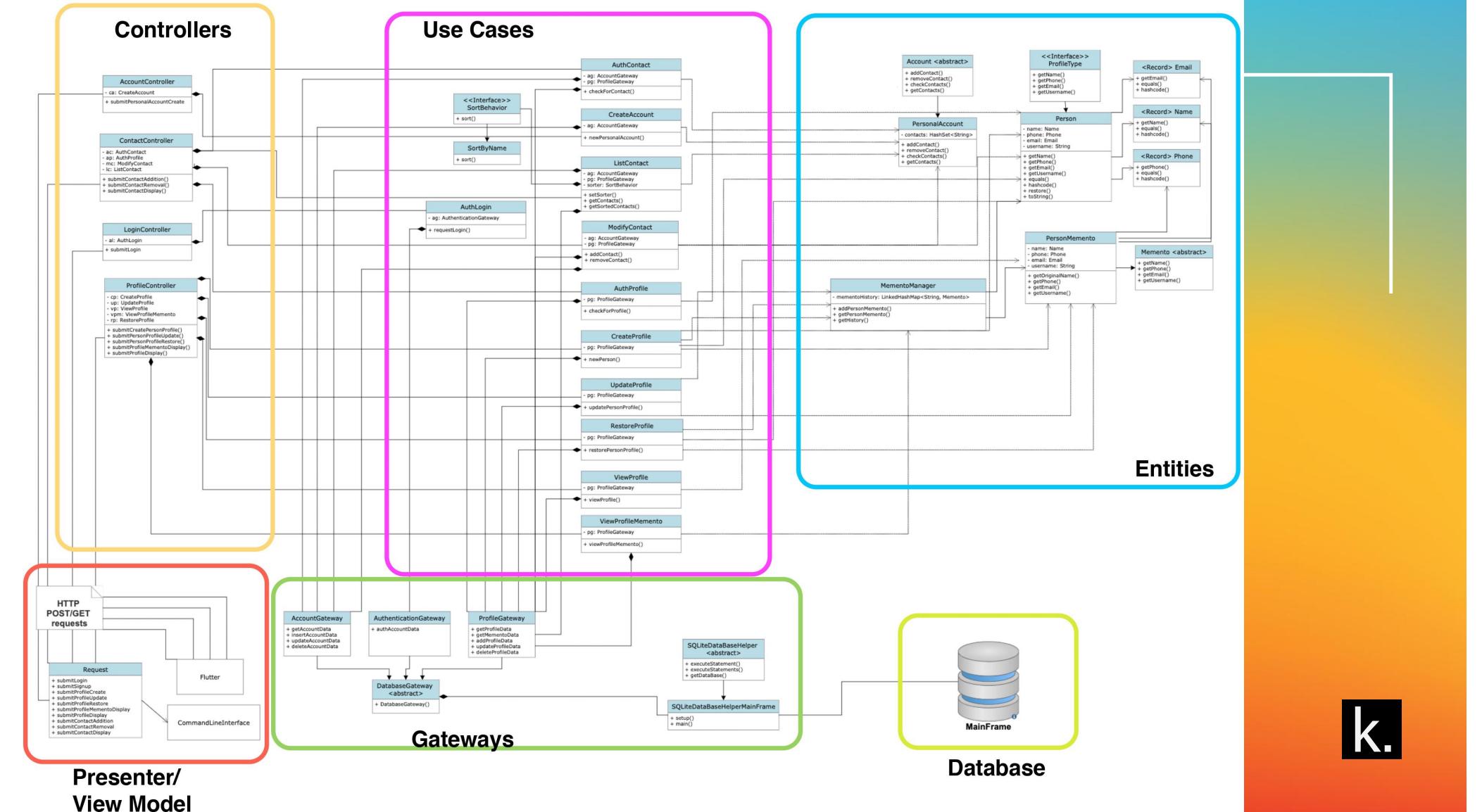
5. Kard's Backend

The brains of the app

k.



k.



k.

Design Patterns

Command

Memento

Strategy



Command – SQLite Database

Represent an action performed by a system as a class.

Allows for more flexibility in the operation being performed.

Decouples the description of an operation from the caller and the fulfiller.



Memento - Profiles

This design pattern enables the program to store snapshots of the previous states of Profiles without breaking the encapsulation principle.

This allows to user to "undo" their changes to their profiles, as they can revert the profiles back to any previous state.

"Originator" *MementoManager* class creates and stores *PersonMemento* objects in a private Hashmap *mementoHistory* from which we can then restore dated snapshots.

Other classes can only access the certain characteristics such as Name, Phone and Email of the Profile at the time of the snapshot, but not the Memento itself.



Strategy – Contact Sorting

Choose from different sorting algorithms at runtime

SortBehavior interface

SortByName implements SortBehavior



6. Kard's Database

k.

Database: Gateways

The gateways are the layer of interaction between Kard's database and the rest of the application.

Obfuscate the Implementation of the database.
Designed to be "transparent" to the usecases.



Database: Helper

Provides an interface to the database for the gateways.

Assists in opening/closing the gateways.

Provides functionality for querying/executing statements on the database.



Database: Query/Statement

Classes following the command design pattern.

Represent operations to be performed on the database.

Instantiated by the gateways to be executed by the helpers.



Database: Manager/SQLiteDatabase

Manager ensures unique instances of databases.

SQLiteDatabase, wrapper class for an SQLite database.

Representation only, the handling of the database is the responsibility of the helpers.



Thanks For Listening!

Any questions?

