

Build a base

Business vision

The Business Model Canvas

Key Partners We are acquiring the basic needs for designing our CRUB from the working files we downloaded from IMDB DB.	Key Activities Our distribution channel is GitHub, where we can have a relationship with the users who download our program. Key Resources The files we have received from IMDB to design our program from.	Value Proposition We are offering a free database to our users.	Customer Relationships We wish to have a relationship of cooperation with our users. They give us feedback which we can use to improve our program. Channels Through GitHub where they can download the program and report any problems to us.	Customer Segments We are looking to make a program for our customers and users that are currently Administrators. The program will be a place where they can CRUB.
Cost Structure Our software is open source and open to a community of developers that can freely help and work on the program in their own free time.			Revenue Streams So far, the service is free, as this is an open source project.	

Source: www.fiaipdonna.it/wp-content/.../04/2013-Business-Model-Canvas-Template.docx

Usecases

UC 1 - Read (Fully dressed)

Name: Build a base

Scope: The program

Primary actor: User

Secondary actor: Administrator

- The person who add new people/movies.

Stakeholder and interests:

- KEA administration.

Pre-conditions:

- User being able to open/use/close program,
- The CRUD operations (Create, Read, Update, Delete)

Succes Guarantee:

Main success scenario:

1. The program takes user input requesting information on a specific person.
2. The program searches through the database, and locates the person based off the input.
3. The program offers a list of people, except those which are marked deleted, with matching data and highlights the ID.
4. The user exits the program via the menu.

Extensions:

1. The program crashes due to hardware and/or software failure.
2. The search has no matches.

Special Requirements:

1. The program can manage large quantities of data.
2. The program needs to be accessible without internet.(?)
3. The program should have “easy to use” search functionality:
You don't need to be an expert to find people/movies etc.

UC 2 - Create (Brief)

Main success scenario:

The user starts the program and requests to add a new Movie. The Movie is added at the bottom, so no other data needs to be rearranged to make space for it, and no indexes, if created, need updating.

UC 3 - Update (Brief)

Main success scenario

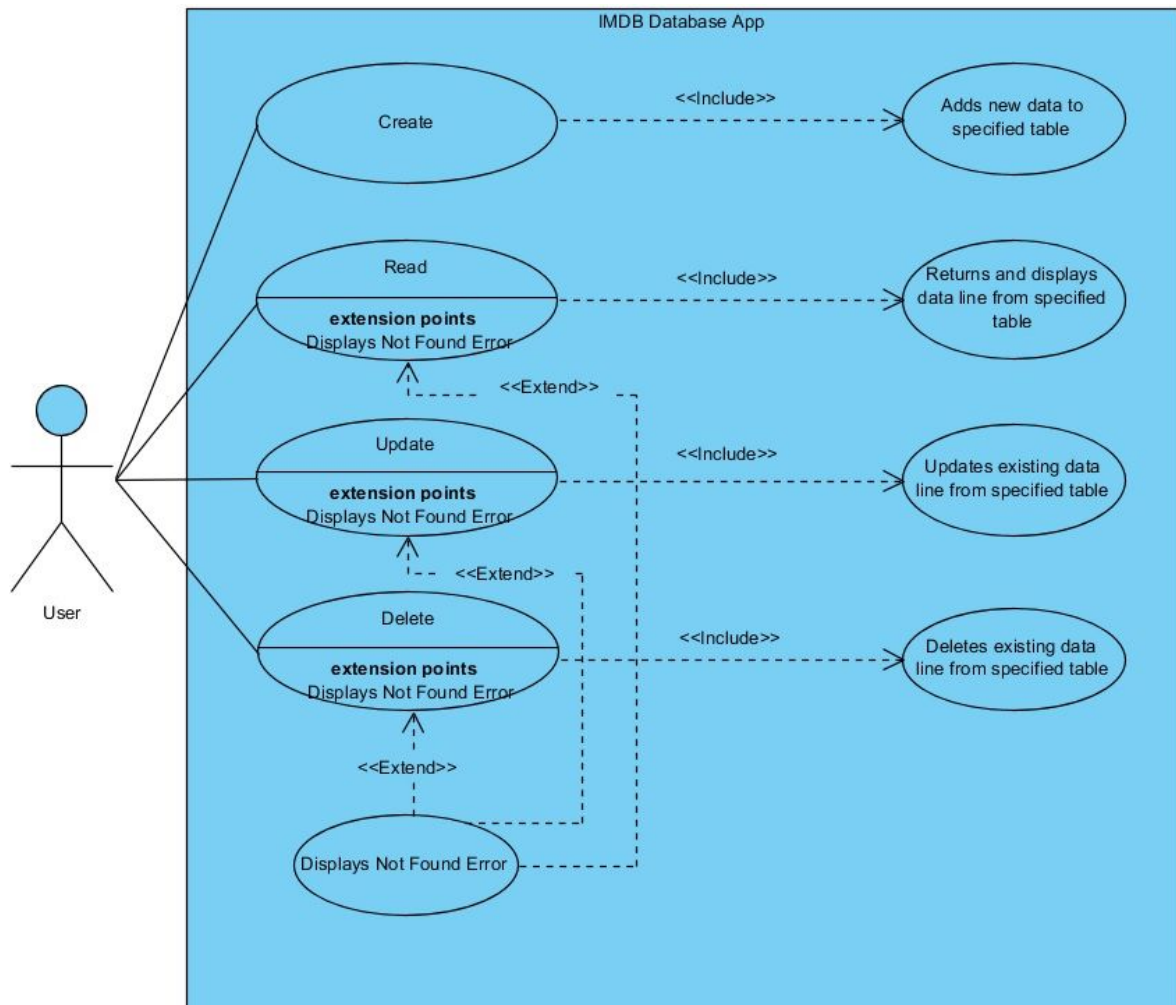
The user starts the program and requests to update the number of episodes in a season, the program performs the necessary updates. The program prevents users from updating IDs. Deleted entries cannot be updated unless explicitly undeleted.

UC 4 - Delete (Brief)

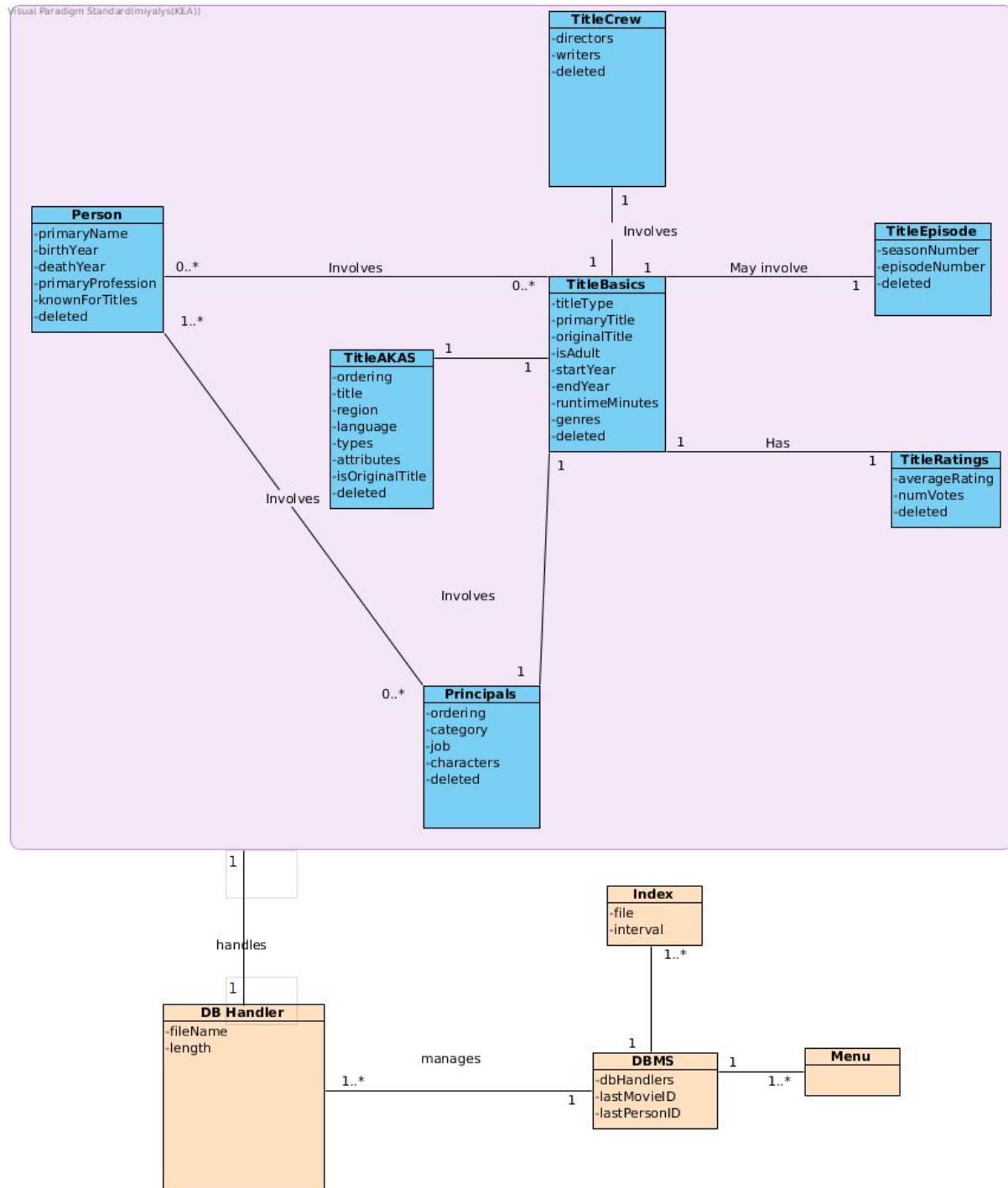
Main success scenario:

The user starts the program and requests to delete/deactivate/remove existing entries. Deleted entries remain in the database but are marked deleted.

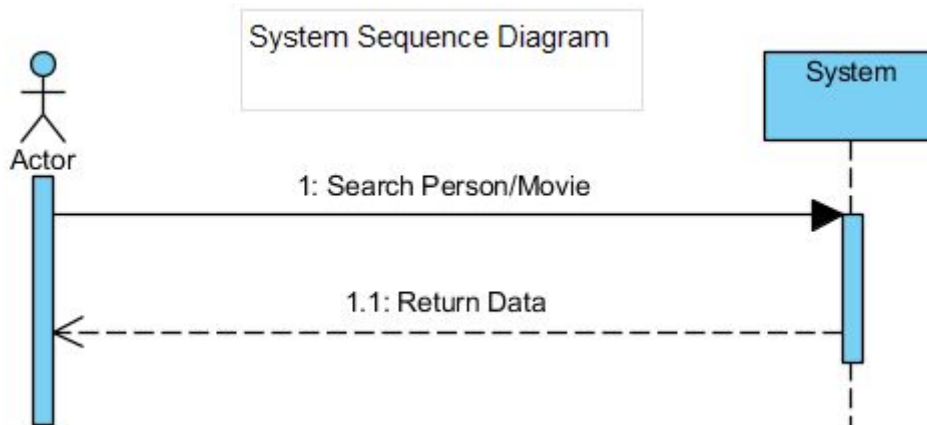
Use Case Diagram



Domain model

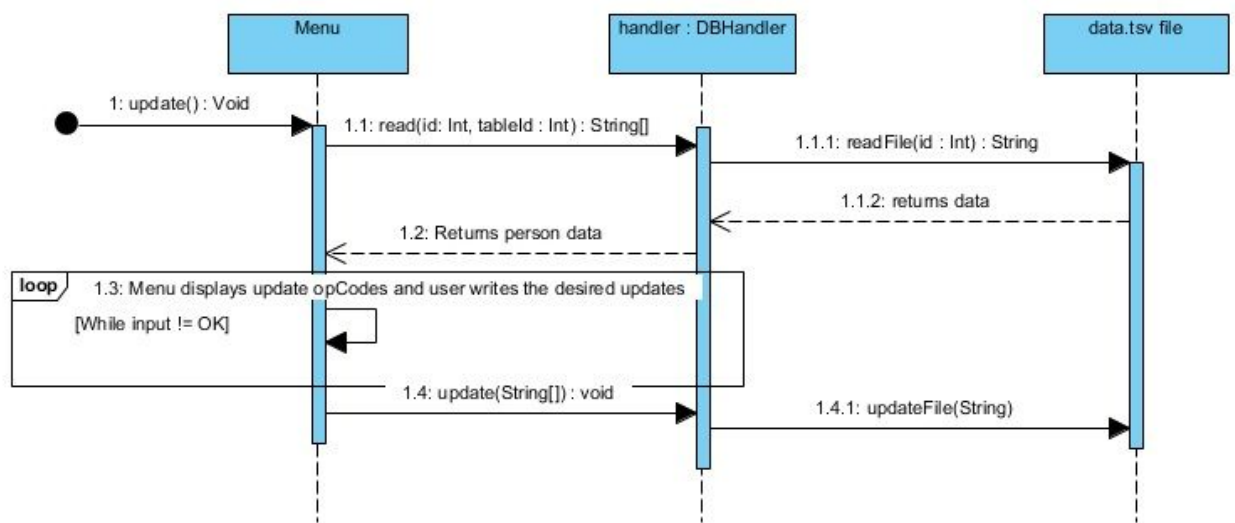


System Sequence Diagram



Sequence Diagram

UPDATE:



Class Diagram

