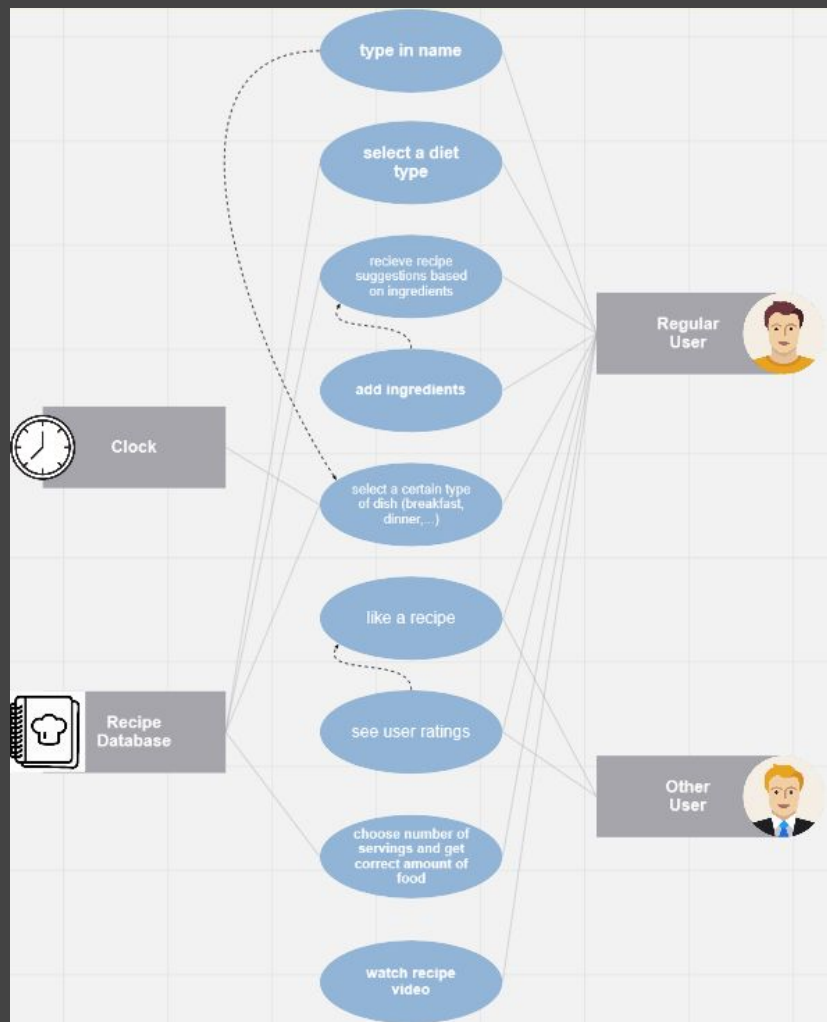


# Team Pacman



- Use Cases, Goals & Design Principles -

# UML USE CASES



## ... use cases in detail

Use case name	Select a diet type
Related requirements	The user wants to choose certain kinds of food that he wants not be included in the recipe recommendations
Goal in context	To get recipes based on this selection
Preconditions	App must be started, name must be given in, options must be shown
Successful end condition	recipes are shown based on the chosen diet type
Failed end condition	all kinds of recipes are shown even if a certain diet type has been selected
Actors	Regular user, recipe database
Trigger	selection screen is shown
Included cases	none
Main flow	<ol style="list-style-type: none"><li>1. button related to specific diet type is clicked</li><li>2. selection is somehow marked</li><li>3. action is committed with a button</li></ol>

Use case name	Add ingredients
Related requirements	The user wants to get recipes based on the ingredients he has at home
Goal in context	Collecting all the ingredients the user wants to use
Preconditions	App must be started, options must be shown, input fields available
Successful end condition	all necessary ingredients are added
Failed end condition	no ingredients are added, wrong ingredients are added
Actors	Regular user
Trigger	input field / selection options are shown
Included cases	none
Main flow	<ol style="list-style-type: none"><li>1. button to add ingredients is clicked</li><li>2. relevant ingredients are selected by adding or typing in</li><li>3. action is committed with a button</li></ol>



## ... use cases in detail

Use case name	Type in name
Related requirements	The user wants to get individual recommendations
Goal in context	Achieve a personal connection to the user
Preconditions	App must be started, "Let's start" button must be clicked
Successful end condition	name is created, user will be individually welcomed
Failed end condition	name could not be created, no name is typed in
Actors	Regular user
Trigger	input field is shown
Included cases	none
Main flow	<ol style="list-style-type: none"><li>1. input field is clicked</li><li>2. name must be typed in</li><li>3. action is committed with a button</li></ol>

Use case name	Recieve recipe suggestions based on ingredients
Related requirements	The user wants to get recipe recommendations based on the ingredients that he has at home / that he has selected
Goal in context	Give recipe suggestions that the user likes to cook
Preconditions	App must be started, ingredients must be added
Successful end condition	suitable recipes are suggested
Failed end condition	nonsuitable recipes are suggested, no recipe is suggested
Actors	Regular user, recipe data base
Trigger	Button to suggest the dishes is clicked
Included cases	add ingredients
Main flow	<ol style="list-style-type: none"><li>1. ingredients are added and button to suggest dishes is clicked</li><li>2. recipes are loading</li><li>3. list of recipes is shown</li><li>4. recipes can be selected by clicking</li></ol>



## ... use cases in detail

Use case name	Select a certain type of dish
Related requirements	The user wants to get choose between different kinds of dishes depending on the time of the day
Goal in context	To automatically suggest dishes for breakfast, dinner, lunch at the related time of the day
Preconditions	App must be started, main screen must be shown, recipe suggestion is launched, app is connected with clock system
Successful end condition	recipes related to actual time of the day are shown
Failed end condition	all kinds/wrong kind of dishes are suggested, no dishes at all are shown
Actors	Regular user, clock, recipe database
Trigger	time, recipe suggestion is launched
Included cases	none
Main flow	<ol style="list-style-type: none"><li>1. app is opened at a certain time of the day</li><li>2. recipe suggestion is launched and selections are made</li><li>3. recipes are forced to be suggested by clicking on a button</li><li>4. user is welcomed related to the time of the day</li><li>5. tab "dinner/breakfast/lunch" is selected depending on time of the day</li><li>6. specific recipes are shown</li></ol>

Use case name	Like a recipe
Related requirements	The user wants to save the recipe
Goal in context	add recipe to the favorite list and rate is as "good"
Preconditions	App must be started, recipe suggestion flow must be completed, recipe must be opened
Successful end condition	recipe is liked and added to favorite list
Failed end condition	favorite button is not filled, recipe could not be added to favorite list
Actors	Regular user, other user
Trigger	recipe is opened
Included cases	none
Main flow	<ol style="list-style-type: none"><li>1. select a recipe</li><li>2. click on the favorite button</li><li>3. button is marked</li><li>4. recipe can be found by clicking on the favorite list button in the app bar</li></ol>



## ... use cases in detail

Use case name	See user ratings
Related requirements	The user wishes to choose recipes based on the user ratings
Goal in context	show number of likes within recipe description to provide valuable comparisons
Preconditions	App must be started, recipe suggestion flow must be completed, recipe must be opened
Successful end condition	recipe description contains correct number of likes given by other users
Failed end condition	likes are not presented, wrong number is presented
Actors	Regular user, other user
Trigger	recipe is opened
Included cases	like a recipe
Main flow	<ol style="list-style-type: none"><li>1. recipe is selected</li><li>2. number of likes is presented in the upper recipe description next to a favorite-button</li><li>3. number may be compared to other recipes</li></ol>

Use case name	choose number of servings
Related requirements	The user might cook for several people and wants to adjust the number of servings
Goal in context	show correct amount of food related to the selected number of servings
Preconditions	App must be started, recipe suggestion flow must be completed, recipe must be opened
Successful end condition	number of servings is typed in and ingredients are displayed in the right quantity
Failed end condition	quantity does not change in the correct way, quantity does not change at all
Actors	Regular user, recipe database
Trigger	recipe is opened, input field for number of servings is displayed
Included cases	none
Main flow	<ol style="list-style-type: none"><li>1. recipe is selected</li><li>2. number of servings is adjusted by clicking on + or - next to the initial number</li><li>3. correct amount is automatically loaded and displayed next to the related ingredients</li></ol>



## ... use cases in detail

Use case name	watch recipe video
Related requirements	The user wants to see the whole cooking process and get illustrative advice how to prepare the dish
Goal in context	provide an explanatory video that shows the cooking process
Preconditions	App must be started, recipe suggestion flow must be completed, recipe must be opened, device must be able to play videos
Successful end condition	video is running
Failed end condition	video can not be opened, is not running, no video is shown
Actors	Regular user
Trigger	video play button in recipe description
Included cases	none
Main flow	<ol style="list-style-type: none"><li>1. recipe is selected</li><li>2. video is started by clicking "play" button</li><li>3. video is opened full screen</li><li>4. video is running</li></ol>



# Usability goals & design principles





Visibility of the  
system status  
(feedback, state  
communication)

Match  
between  
system & real  
world (natural  
language)

User control &  
freedom  
(undo/redo,  
clear way to  
exit)

Consistency &  
standards (use  
same  
buttons/design  
for same things)

Error  
prevention  
(warnings,  
undo etc.)


Recognition rather  
than recall (reduce  
info that user has to  
remember, no long  
tutorials, help in  
context)

Flexibility &  
efficiency of use  
(shortcuts,  
personalization,  
customization)

Aesthetic and  
minimalist design  
(no irrelevant info,  
no distractions,  
focus on relevant  
stuff)

Help users recognize,  
diagnose and recover  
from errors (use  
traditional error  
messages, tell what  
went wrong, solution)

Help &  
documentation  
(easy to search,  
placed at the right  
moment, concrete  
steps)

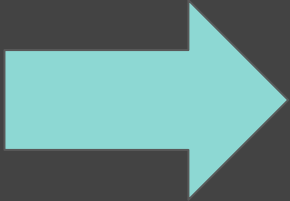
 most relevant ones for  
us

## Usability Requirements from Personas/Scenarios:

- get relevant recipes
- filter ingredients
- quick and easy use of the app, time-saving
- nice picture
- see all relevant things at a glance
- intuitive

## Usability Goals

- create an intuitive, consistent and illustrative design
- avoid irrelevant information
- give the user the ability to personalize the information
- provide simple communication, which is easy to understand
- provide solutions to errors to safe time



## Improvements

- create an improved user flow that is understandable for the users
- use templates for buttons which functions are well trained from other apps
- provide button descriptions
- provide settings that can be changed at all times
- provide feedback regarding the system status (show when app is loading or if sth went wrong)

