Dokumentacja projektu TODO

Wykonali:

Maciej Jaworowski

Maciej Bąk

Gr. ISI1 sem 7

# Cel projektu

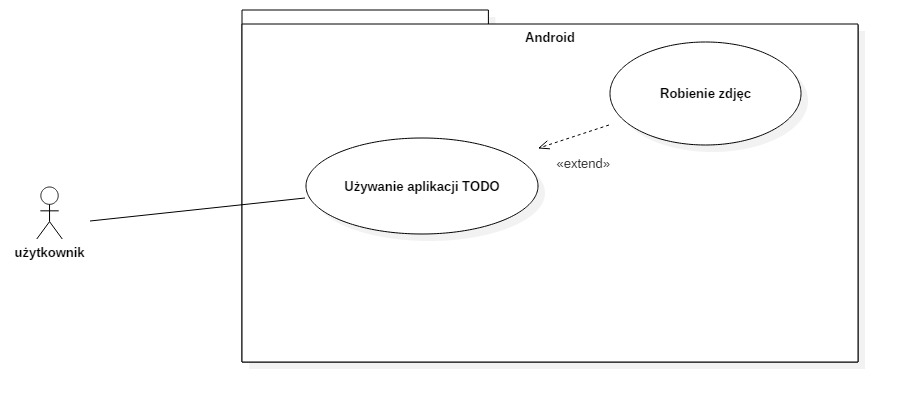
Celem projektu jest stworzenie aplikacji mobilnej usprawniającej zarządzaniem i planowaniem zadań do zrobienia tj. aplikacji typu TODO. Aplikacja pozwala na zarządzanie zadaniami wraz z ich priorytetami.

Projekt został stworzony do pracy na systemie Android w wersji 7.0 API 24 lub nowszej.

# Lista funkcjonalności

* Wyświetlanie zadań wraz z priorytetami
* Dodawanie zadań do listy
* Oznaczanie zadań jako zrobione
* Edytowanie zadań
* Usuwanie zadań
* Sortowanie zadań po priorytecie oraz nazwie
* Dodawanie załączników (zdjęć)
* Usuwanie załączników
* Wyświetlanie listy załączników przypisanych do zadania

# Diagram przypadków użycia



Użytkownik korzysta z aplikacji TODO, która jest umieszczona w systemie Androida. Aplikacja jest rozszerzona o możliwość robienia zdjęć oraz ich załączania. Funkcjonalność ta jest realizowana przez system operacyjny Androida.

# Lista Klas z funkcjami

Klasa: AddToDoElement

Dziedziczy po: AppCompatActivity

|  |
| --- |
| Funkcja |
| onCreate(savedInstanceState: Bundle?) |

Klasa: Attachments

Dziedziczy po: -

|  |
| --- |
| Funkcja |
| constructor(id: Int, taskID: Int, dateAdded: Int, filePath: String) |

Klasa: DatabaseHandler

Dziedziczy po: SQLiteOpenHelper

|  |
| --- |
| Funkcja |
| onCreate(db: SQLiteDatabase?) |
| onUpgrade(db: SQLiteDatabase?, oldVersion: Int, newVersion: Int) |
| insertData(toDoList: ToDoList) |
| readDataDone(): MutableList<ToDoList> |
| readDataNotDone(): MutableList<ToDoList> |
| readOneObjectToDO(id: Int): ToDoList |
| updatePriority(id: Int, priority: String) |
| updateDone(id: Int, done: Int) |
| updateToDo(id: Int, todo: String) |
| deleteToDO(id: Int) |
| insertAttachment(filepath: String, taskID: Int) |
| readDataAttachments(taskID: Int): MutableList<Attachments> |
| deleteAttachments(list: MutableList<Int>): Int |
|  |

Klasa: EditToDoElement

Dziedziczy po: AppCompatActivity

|  |
| --- |
| Funkcja |
| onCreate(savedInstanceState: Bundle?) |

Klasa: MainActivity

Dziedziczy po: AppCompatActivity

|  |
| --- |
| Funkcja |
| onCreate(savedInstanceState: Bundle?) |
| onCreateOptionsMenu(menu: Menu): Boolean |
| sort(option: CharSequence, viewDone: Boolean) |
| openSortOptions(): Boolean |
| fun onOptionsItemSelected(item: MenuItem): Boolean |
| sort(option: CharSequence): Boolean |

Klasa: MainAdapter

Dziedziczy po: RecyclerView.Adapter<CustomViewHolder>()

|  |
| --- |
| Funkcja |
| fun getItemCount(): Int |
| sortByName(desc: Boolean = false) |
| sortByPriority(desc: Boolean = false) |
| fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): CustomViewHolder |
| onBindViewHolder(holder: CustomViewHolder?, position: Int) |
|  |

Klasa: CustomViewHolder

Dziedziczy po: RecyclerView.ViewHolder(view)

Klasa: MainAdapterDone

Dziedziczy po: RecyclerView.Adapter<CustomViewHolderDone>()

|  |
| --- |
| Funkcja |
| fun getItemCount(): Int |
| fun onCreateViewHolder(parent: ViewGroup?, viewType: Int): CustomViewHolder |
| onBindViewHolder(holder: CustomViewHolder?, position: Int) |
|  |

Klasa: CustomViewHolder

Dziedziczy po: RecyclerView.ViewHolder(view)

Klasa: ToDoDetail

Dziedziczy po: AppCompatActivity

|  |
| --- |
| Funkcja |
| onCreate(savedInstanceState: Bundle?) |
| listAttachments() |
| saveAttachment(option: CharSequence) |
| requestPermission() |
| onRequestPermissionsResult(requestCode: Int, permissions: Array<String>, grantResults: IntArray) |
| onActivityResult(requestCode: Int, resultCode: Int, data: Intent) |
| openCamera() |
| updatePriority(ToDoId: Int) |
| checkPriority(priority: String): Array<out String>? |
|  |

Klasa: ToDoList

Dziedziczy po: -

|  |
| --- |
| Funkcja |
| constructor(toDo: String, done: Int, priority: String) |
| constructor() |

# Wybrany algorytm

Po kliknięciu menu opcji sortowania zostaje wyświetlone okno dialogowe z opcjami sortowania. Po wybraniu przez użytkownika konkretnej opcji sortowania zostaje przeładowany ekran główny z posortowanym elementami.

