**Państwowa Wyższa Szkoła Informatyki i Przedsiębiorczości w Łomży**



Przedmiot:   
**Projekt zespołowy II**

Prowadzący:

**Janusz Rafałko**

Projekt wykonali:

**Daniel Dudziak, Kiernozek Michał,   
Rafał Ślesicki, Krzysztof Wysocki  
Informatyka, niestacjonarne, semestr VI**

|  |  |
| --- | --- |
| Wersja dokumentacji numer: | 10 |

**Wstęp**

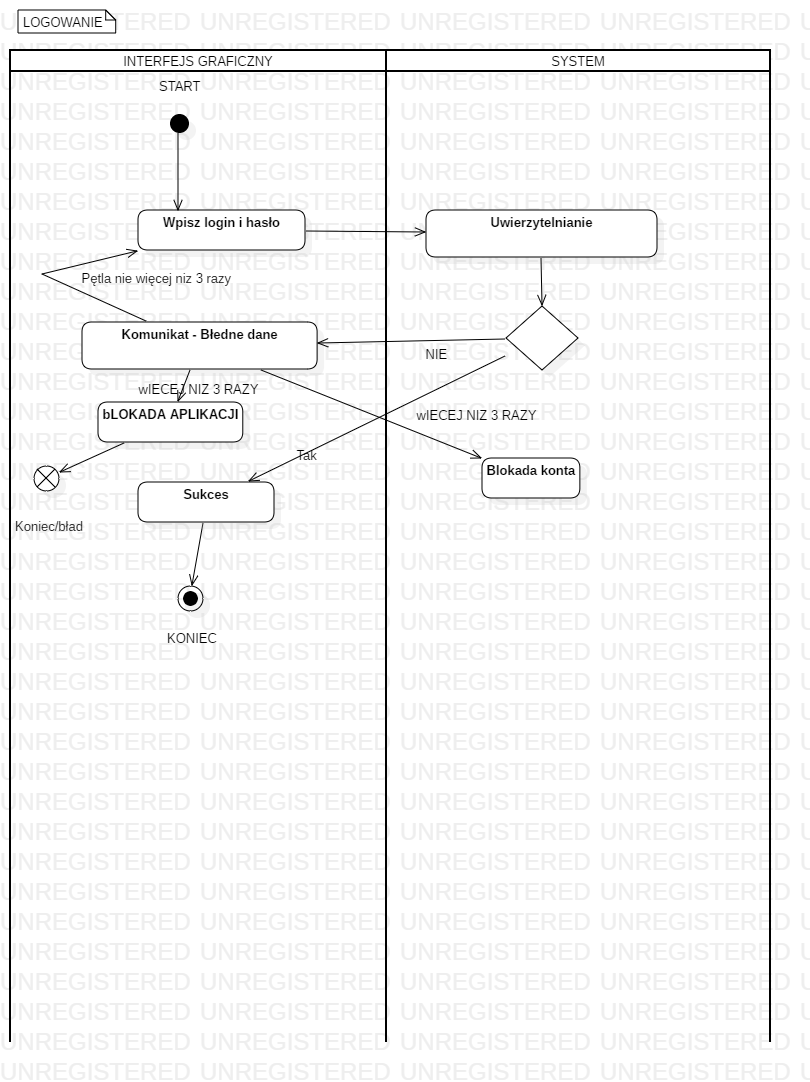
Szybki rozwój techniki oraz postępująca miniaturyzacja sprawiły, że coraz powszechniejsze stało się posiadanie smartfonów. Miniaturowe urządzenia, mieszczące się w kieszeni, coraz bardziej przypominają swoją funkcjonalnością komputer. Popularyzacja tego typu urządzeń pociągnęła za sobą również rozwój systemów na platformy mobilne.

Poruszonym zagadnieniem w opracowanej dokumentacji jest tworzenie oprogramowania na system Android. Wydany kilka lat temu system stworzony przez Google, bardzo szybko zdobył popularność i jest obecnie wiodącą platformą na nowoczesne telefony komórkowe. Udostępniony zestaw narzędzi dla programistów oraz obszerna dokumentacja znacznie ułatwiają tworzenie oprogramowania na nowy system. Dzięki udostępnionemu API (ang. Application Programming Interface) można korzystać z wielu gotowych funkcji, które ułatwiają na przykład korzystanie z wbudowanego w telefon sprzętu. Rozwój miniaturyzacji pozwolił również na umieszczenie w telefonie wielu urządzeń rozszerzających jego funkcjonalność, jak na przykład odbiornika GPS (ang. Global Positioning System), żyroskopu, modułu Wi-Fi czy miniaturowej kamery.

**Opis**

Aplikacja Foods App to znakomicie zaprojektowana interaktywna książka kucharska, w której można znaleźć znakomite przepisy na każdą okazję. Aplikacja jest pełna podzielonych na kategorie przepisów (dania główne, desery). Oferuje przepisy opatrzone zdjęciami jak również linki do filmów i stron źródłowych. Przepisy w aplikacji dodawane są przez użytkowników serwisu www.themealdb.com, co gwarantuje, że bez problemu będziemy mogli przygotować je samodzielnie we własnej kuchni

Tutaj powinien być diagram UML.



**Wymagania niefunkcjonalne użytkownika**

1. **W obszarze funkcjonalności:**
2. Logo firmy umieszczone jest w lewym górnym rogu.
3. Kolory aplikacji to: biały, bordowy, szary.
4. Aplikacja działa na systemach Android nie starszych niż 8.
5. **W obszarze użyteczności:**
6. Aplikacja używa czcionki 11 pikseli przy rozdzielczości x na x.
7. **W obszarze niezawodności:**
8. Aplikacja będzie dostępna dla użytkowników w ciągu średnio 99% czasu w każdym roku.
9. Aplikacja będzie dostępna 24/7/365 średnio 99% czasu.
10. **W obszarze wydajności:**
11. Aplikacja będzie umożliwiać korzystanie z niej przez 1200 osób jednocześnie, Osób, które są umiejscowione w różnych lokalizacjach.
12. Struktura aplikacji w raz z danymi 10 000 000 rekordów nie będzie większa niż 5TB. Same dane rekordów bez załączników.
13. **W obszarze wsparcia:**
14. Wszystkie błędy krytyczne aplikacji zostaną naprawione w ciągu 48 h roboczych.
15. Wszyscy użytkownicy aplikacji mogą zgłaszać błędy na dedykowany adres email i dedykowany telefon
16. Wszystkie błędy aplikacji będą monitorowane. Monitorowany będzie sposób i czas ich naprawy.
17. Do aplikacji zgłaszania błędów będą miały dostęp wszystkie zarejestrowane osoby.
18. **Narzędzia programistyczne  
      
    Repozytorium**

Repozytorium zostało stworzone w witrynie internetowej „**GitHub**”.

1. **Środowisko programistyczne**

**ANDROID STUDIO** – środowisko to tworzenia aplikacji w języku Kotlin

**Przykładowe zrzuty kodu aplikacji:**

package com.haerul.foodsapp.adapter;  
  
import android.content.Context;  
import android.support.annotation.NonNull;  
import android.support.v7.widget.RecyclerView;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ImageView;  
import android.widget.TextView;  
  
import com.haerul.foodsapp.R;  
import com.haerul.foodsapp.model.Categories;  
import com.squareup.picasso.Picasso;  
  
import java.util.List;  
  
import butterknife.BindView;  
import butterknife.ButterKnife;  
  
public class RecyclerViewHomeAdapter extends RecyclerView.Adapter<RecyclerViewHomeAdapter.RecyclerViewHolder> {  
  
 private List<Categories.Category> categories;  
 private Context context;  
 private static ClickListener *clickListener*;  
  
 public RecyclerViewHomeAdapter(List<Categories.Category> categories, Context context) {  
 this.categories = categories;  
 this.context = context;  
 }  
  
 @NonNull  
 @Override  
 public RecyclerViewHomeAdapter.RecyclerViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {  
 View view = LayoutInflater.*from*(context).inflate(R.layout.*item\_recycler\_category*,  
 viewGroup, false);  
 return new RecyclerViewHolder(view);  
 }  
  
 @Override  
 public void onBindViewHolder(@NonNull RecyclerViewHomeAdapter.RecyclerViewHolder viewHolder, int i) {  
  
 String strCategoryThum = categories.get(i).getStrCategoryThumb();  
 Picasso.*get*().load(strCategoryThum).placeholder(R.drawable.*ic\_circle*).into(viewHolder.categoryThumb);  
  
 String strCategoryName = categories.get(i).getStrCategory();  
 viewHolder.categoryName.setText(strCategoryName);  
 }  
  
  
 @Override  
 public int getItemCount() {  
 return categories.size();  
 }  
  
 static class RecyclerViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {  
 @BindView(R.id.*categoryThumb*)  
 ImageView categoryThumb;  
 @BindView(R.id.*categoryName*)  
 TextView categoryName;  
 RecyclerViewHolder(@NonNull View itemView) {  
 super(itemView);  
 ButterKnife.*bind*(this, itemView);  
 itemView.setOnClickListener(this);  
 }  
  
 @Override  
 public void onClick(View v) {  
 *clickListener*.onClick(v, getAdapterPosition());  
 }  
 }  
  
  
 public void setOnItemClickListener(ClickListener clickListener) {  
 RecyclerViewHomeAdapter.*clickListener* = clickListener;  
 }  
  
  
 public interface ClickListener {  
 void onClick(View view, int position);  
 }  
}

**Harmonogram:**

|  |  |  |
| --- | --- | --- |
| **1** | **Plan projektu** | **- temat**  **- wstępne założenia**  **-wstępne funkcjonalności**  **- narzędzia programistyczne** |
| **2** | **Dokumentacja wstępna** | **- cel projektu**  **- opis**  **- diagram przypadków użycia**  **- wymagania funkcjonalne/niefunkcjonalne użytkownika** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Podział prac:**

Dokumentacja jest dokumentacją wstępną. Wraz z rozwojem projektu będzie rozwijana/modyfikowana.