

Projekt w grupie programistycznej

Organizacja zajęć uczelnianych

Grupa 3

1. Podział ról:

- 1.1. Project manager - Patryk 12525
- 1.2. Product owner - Paulina 13817
- 1.3. Developer - Jakub 13862
- 1.4. UX designer - Patryk 14943
- 1.5. Full Stack Developer – Emil 15068

2. Opis aplikacji

2.1. Cel aplikacji

- Lepsza organizacja czasowa użytkowników,
- Uproszczenie i lepsza organizacja komunikacji między dziekanatem, prowadzącym zajęcia i studentem,
- Lepsza organizacja sprzętu i sal przed zajęciami,

2.2. Założenia

- Plan zajęć (kalendarz) dla studenta oraz prowadzącego wraz z mapą dojazdu (i.e. Google Maps) i informacja o czasie niezbędnym na dojazd,
- Moduł wykładowcy wyposażony o podgląd wszystkich grup studentów, które prowadzi,
- Moduł studenta bez widoku elementów dokumentacyjnych widocznych w interfejsie dla wykładowcy,
- Interfejs dla dziekanatu informujący m.in. o kolizjach w planie zajęć,
- Serwerownia na terenie Lublina,
- Budżet wstępny: 600 000 zł (docelowy 1 000 000 zł),
- Udostępnienie prototypu aplikacji: 6 miesięcy od daty rozpoczęcia prac,
- Udostępnienie pełnej wersji aplikacji: 10 miesięcy od daty rozpoczęcia prac,

2.3. Użytkownicy aplikacji

- Studenci oraz pracownicy uczelni wyższej (prowadzący zajęcia, pracownicy dziekanatu),
- Aplikacja będzie użytkowana maksymalnie na terenie Polski, najczęściej na terenie Lublina (z perspektywą ekspansji na pozostałe miasta na terenie Polski). Ilość użytkowników powinna zamykać się w granicach 1500 osób.

2.4. Wykorzystanie produktu

- Aplikacja napisana dla systemu Android (aplikacja telefoniczna – z możliwością rozszerzenia w przyszłości),
- Nieadministracyjny interfejs główny dla studentów oraz prowadzących,
- Administracyjny Interfejs dla dziekanatu,
- Ładowanie aplikacji maksymalnie do 5 sekund (preferowane 3,4 sekundy),
- Działanie aplikacji w ramach roku akademickiego (zjazdy stacjonarne i niestacjonarne),
- Preferowany termin konserwacji w trakcie przerw w funkcjonowaniu uczelni (przerwy świąteczne, ferie, weekendy bez zjazdów),

2.5. Potrzeby użytkowników

- Poprawnie stworzona aplikacja będzie pomagała studentom i prowadzącym dotrzeć na miejsce prowadzenia zajęć,
- Ułatwienie przemieszczania się pomiędzy budynkami– gdy zajęcia prowadzone są w różnych lokalizacjach,
- Wyświetlanie najważniejszych komunikatów dla wybranych grup użytkowników,
- W przypadku odwołania zajęć: studenci oraz prowadzący otrzymają komunikat w aplikacji oraz zostanie wysłany mail z potwierdzeniem odwołania zajęć na skrzynki mailowe,
- Aplikacja nie będzie zintegrowana z żadnym istniejącym produktem, ale powinna mieć możliwość wysyłania komunikatów (powiadomień) na skrzynki mailowe,
- Każdy z użytkowników będzie posiadał własne konto w aplikacji w podziale na profile użytkowników: prowadzący, student, pracownik dziekanatu,
- Sposób logowania: wpisanie loginu uczelnianego (adresu e-mail) oraz hasła. (aplikacja będzie miała dostęp do bazy danych z odpowiednimi danymi tak, aby aplikacja weryfikowała użytkowników automatycznie oraz dostosowywała profile i uprawnienia do odpowiednich statusów użytkowników,
- Na stronie głównej powinny znajdować się informacje o zbliżających się zajęciach, po kliknięciu na dane zajęcia rozwijają się opcje dojazdu (po kliknięciu pokazuje się mapa dojazdu) oraz szczegóły (po kliknięciu znajdują się szczegóły zajęć, czas na dojazd oraz czas przerwy pomiędzy poprzedzającymi zajęciami),

2.6. Interfejs i opis

- **Wygląd aplikacji**

- **Pasek górny**

zawiera tytuł aktualnej zakładki oraz przycisk ustawień, który po przejściu do tej zakładki zmienia kolor na czerwony.



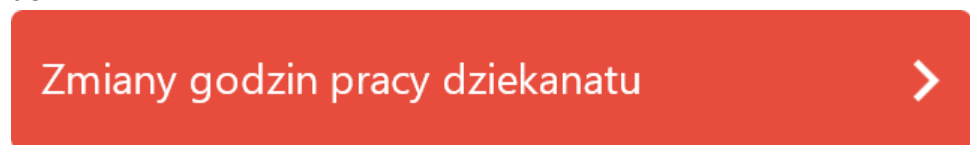
- **Pasek dolny**

Pozwala przemieszczać się pomiędzy zakładkami poprzez wybranie ikony. Ikona aktualnie wyświetlanej zakładki zmienia kolor na czerwony



- **Strona główna**

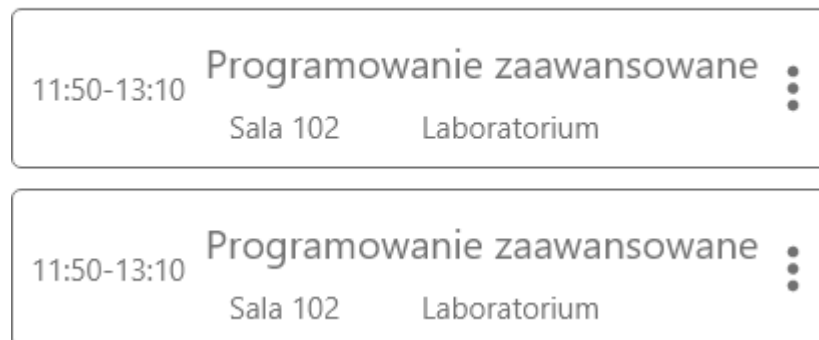
Pod paskiem górnym wyświetla nieprzeczytane powiadomienia na czerwonym tle.



Po kliknięciu w to pole przechodzimy do panelu powiadomień.

W centrum strony znajduje się lista nadchodzących zajęć w danym dniu

Nadchodzące zajęcia



Wyświetlane są takie informacje jak: nazwa przedmiotu, godziny rozpoczęcia i zakończenia, sala oraz rodzaj zajęć.

Klikając jedno z pól, uzyskamy szczegółowe informacje na temat zajęć, oraz mamy możliwość ustawienia powiadomienia.

Programowanie zaawansowane

Laboratorium

Godziny zajęć: 11:50-13:10

Ustaw przypomnienie:



Prowadzący zajęcia: Dr inż. Jan Kowalski

Sala: 102


- **Kalendarz**

W lewym górnym rogu znajduje się pole do wyboru miesiąca i roku.

Sty 2023 ▼

W Środkowej części strony wyświetla się kalendarz. Aktualny dzień zaznaczony jest czerwonym polem

N	P	W	Ś	C	P	S
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11



Pod kalendarzem znajduje się plan na wybrany dzień.

17.01.2023

11:50-13:10 Programowanie zaawansowane
Sala 102 Laboratorium

11:50-13:10 Programowanie zaawansowane
Sala 102 Laboratorium

- **Mapa**

Na całej stronie wyświetla się mapa. W górnej części znajduje się pole wyszukiwania.



W dolnym prawym rogu umieszczona jest ikonka odpowiadająca za lokalizację gps.



- **Powiadomienia**

Na górze znajduje się pasek wyszukiwania oraz pole wyboru filtrów.



Filtry

Pod polem wyszukiwania wyświetla się historia powiadomień.

Zmiany godzin pracy dziekanatu

05.01.2023

Lorem ipsum dolor sit amet, consectetur

Lorem ipsum dolor sit amet, consectetur

Zmiany godzin pracy dziekanatu

05.01.2023

Lorem ipsum dolor sit amet, consectetur

Lorem ipsum dolor sit amet, consectetur

Każde powiadomienie zawiera tytuł, opis oraz datę otrzymania.

- **Ustawienia**

Po przejściu do ustawień wyświetli się lista możliwych konfiguracji.

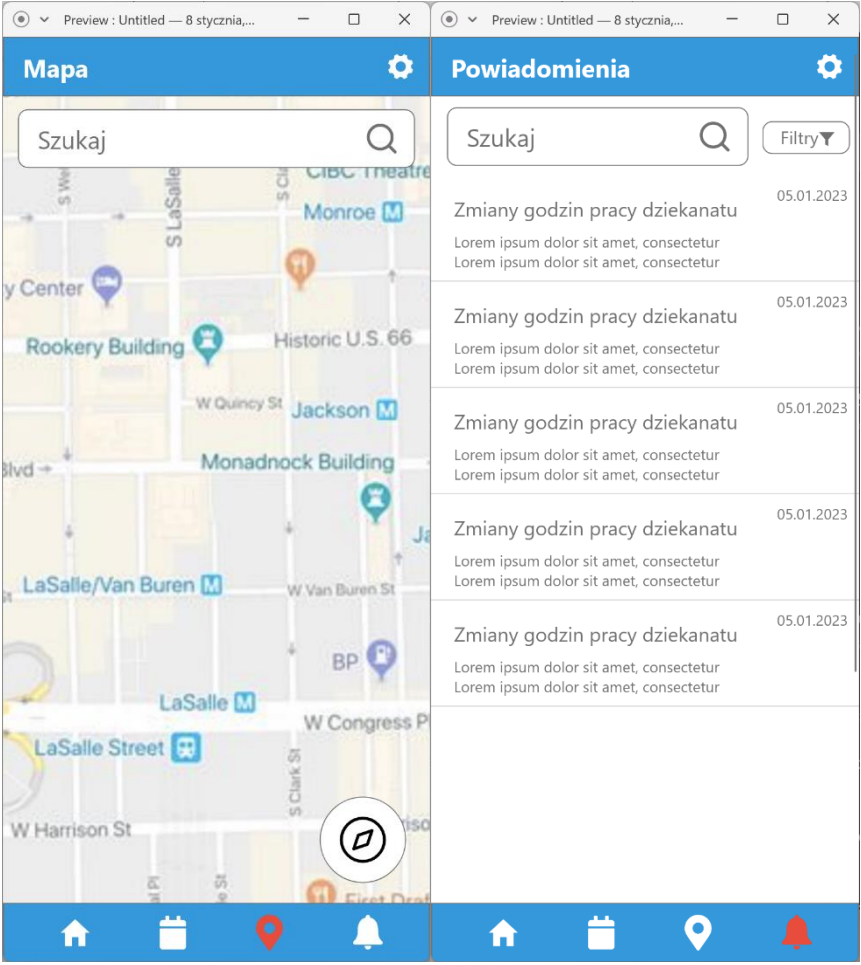
Ustawienia konta

Powiadomienia

Motyw

Pomoc i opinie

Informacje



Preview : Untitled — 8 stycznia,...

Ustawienia

Ustawienia konta

Powiadomienia

Motyw

Pomoc i opinie

Informacje

Preview : Untitled — 8 stycznia,...

Kalendarz

Sty 2023

N	P	W	Ś	C	P	S
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4
5	6	7	8	9	10	11

17.01.2023

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

🏠

📅

📍

🔔

🏠

📅

📍

🔔

Strona główna

← Szczegóły

Zmiany godzin pracy dziekanatu

Nadchodzące zajęcia

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

11:50-13:10

Programowanie zaawansowane

Sala 102Laboratorium

Programowanie zaawansowane

Laboratorium

Godziny zajęć: 11:50-13:10

Ustaw przypomnienie:

15

30

60

120

Prowadzący zajęcia: Dr inż. Jan Kowalski

Sala: 102

2.7. Interfejsy użytkowników

- **Student**

Funkcjonalności:

- Informacja o zbliżających się zajęciach wraz z lokalizacją na mapie,
- Kalendarz z zajęciami dla przypisanej grupy (więcej, niż jedna),
- Mapa dojazdu pomiędzy budynkami uczelni (preferowany Google Maps),
- Interfejs z powiadomieniami,
- Opcja alarmu- przypomnienie o zbliżających się zajęciach (w możliwośćią ustawień własnych, takich jak wyłączenie przypomnienia lub ustawienie czasu oraz dźwięku alarmu),

- **Prowadzący zajęcia**

Funkcjonalności:

- Informacja o zbliżających się zajęciach wraz z lokalizacją na mapie,
- Kalendarz z prowadzonymi zajęciami (więcej, niż jedna),
- Mapa dojazdu pomiędzy budynkami uczelni (preferowany Google Maps),
- Kanał komunikacji z dziekanatem,
- Interfejs z powiadomieniami,
- Opcja alarmu- przypomnienie o zbliżających się zajęciach (w możliwośćią ustawień własnych, takich jak wyłączenie przypomnienia lub ustawienie czasu oraz dźwięku alarmu),

- **Dziekanat**

Podsumowanie funkcjonalności:

- Interfejs zarządzający zajęciami – informacja o kolizjach w zajęciach, zbyt krótkimi/zbyt długimi przerwami pomiędzy zajęciami, możliwość odwołania, tworzenia zajęć i zmiany lokalizacji zajęć (po każdej dokonanej zmianie wysłany komunikat użytkownikom, których ta zmiana dotyczy),
- Interfejs zarządzania powiadomieniami – możliwość wysyłania powiadomień wybranym grupom odbiorców,
- Powiadomienia dla dziekanatu o zbliżających się zajęciach,

2.8. Funkcje aplikacji

- **Interfejs z widoku studenta**

- **Zębatka:** umożliwia ustawienia do aplikacji: zmiana trybu wyświetlania nocny/dzienny; włączenie/wyłączenie powiadomień; wybranie jednej z 3 melodii alarmu; włączanie alarmów przypominających o wydarzeniach (15, 30, 60, 120 minut przed rozpoczęciem wydarzenia);
- **Powiadomienia przypominające:** informuje o zmianach najbliższych zajęć;
- **Nadchodzące zajęcia:** po kliknięciu strzałki mamy możliwość sprawdzenia dojazdu do miejsca zajęć- mapa z informacją o czasie dojazdu (pieszo, samochodem, komunikacją), odległości oraz natężeniu ruchu na odcinku;

- **Przycisk szczegóły:** przedstawia czas trwania zajęć, stopień naukowy, imię, nazwisko prowadzącego, ustawienie dla tego jednego zdarzenia powiadomienia w formie alarmu oraz przypomnienia (15, 30, 60, 120 minut);
 - **Przycisk kalendarz:** przenosi do widoku zajęć na najbliższy dzień (nazwa przedmiotów, lokalizacja, prowadzący, godziny trwania zajęć);
 - **Przycisk mapa:** przedstawia mapę Lublina oraz naszą lokalizację, lokalizacje budynków, w których odbywają się w dniu wybranym zajęcia, wyróżnia budynek, gdzie odbędą się najbliższe zajęcia. Przedstawia czas dojazdu z naszej lokalizacji do budynku, gdzie odbędą się najbliższe zajęcia oraz informację, o której godzinie trzeba wyjść z jednego budynku uczelni aby dotrzeć w odpowiednim czasie na zajęcia odbywające się w innej lokalizacji;
 - **Powiadomienia:** wyświetla komunikaty informujące o zmianach na cały dzień czy zajęcia zostały odwołane, przełożone, zmieniono lokalizację, zmiana prowadzącego ze szczegółowym podglądem, na kiedy przełożone, na jaką lokalizację przeniesiono zajęcia, kto będzie prowadził zajęcia po zmianie;
- **Interfejs prowadzącego zajęcia:**
 - **Zębatka:** umożliwia ustawienia do aplikacji: zmiana trybu wyświetlania nocny/dzienny; włączenie/wyłączenie powiadomień; wybranie jednej z 3 melodii alarmu; włączanie alarmów przypominających o wydarzeniach (15, 30, 60, 120 minut przed rozpoczęciem wydarzenia);
 - **Powiadomienia przypominające:** informuje o zmianach najbliższych zajęć;
 - **Nadchodzące zajęcia:** po kliknięciu strzałki mamy możliwość sprawdzenia dojazdu do miejsca zajęć- mapa z informacją o czasie dojazdu (pieszo, samochodem, komunikacją), odległości oraz natężeniu ruchu na odcinku;
 - **Przycisk szczegóły:** przedstawia czas trwania zajęć, stopień naukowy, imię, nazwisko prowadzącego, ustawienie dla tego jednego zdarzenia powiadomienia w formie alarmu oraz przypomnienia (15, 30, 60, 120 minut);
 - **Przycisk kalendarz:** przenosi do widoku zajęć na najbliższy dzień (nazwa przedmiotów, lokalizacja, grupa studentów, godziny trwania zajęć);
 - **Przycisk mapa:** przedstawia mapę Lublina oraz naszą lokalizację, lokalizacje budynków, w których odbywają się w dniu wybranym zajęcia, wyróżnia budynek, gdzie odbędą się najbliższe zajęcia. Przedstawia czas dojazdu z naszej lokalizacji do budynku, gdzie odbędą się najbliższe zajęcia oraz informację, o której godzinie trzeba wyjść z jednego budynku uczelni aby dotrzeć w odpowiednim czasie na zajęcia odbywające się w innej lokalizacji;
 - **Powiadomienia** wyświetla komunikaty informujące o zmianach na cały dzień czy zajęcia zostały odwołane, przełożone, zmieniono lokalizację, zmiana prowadzącego ze szczegółowym podglądem, na kiedy przełożone, na jaką lokalizację przeniesiono zajęcia, kto będzie prowadził zajęcia po zmianie;

3. Harmonogram prac projektowych

Metodyka, którą projekt zamierza używać to **Scrum**

Full Stack Developer = UX Developer/Developer

Miesiąc prac	Tydzień prac	Osoba odpowiedzialna	Rodzaj prac
1	1	Product Owner	Rozpoznanie biznesowe
1	1	Product Owner	Nakreślenie potrzeb biznesu
1	1	Product Owner	Utworzenie kontraktu
1	2	Product Owner/ Project Manager	Przedstawienie możliwości projektowych klientowi (plus korekta)
1	3	Project Manager	Utworzenie początkowych diagramów projektowych
1	3	Project Manager/ UX Designer	Wizualizacja prototypu rozwiązania
1	3	Product Owner/ Project Manager	Spotkanie z biznesem oraz akceptacja projektu
1	4	Project Manager	Przedstawienie projektu poszczególnym deweloperom
1	4	Project Manager	Wybranie drużyny projektowej
2	1	Project Manager	Wybranie oraz implementacja metodyki projektowej
Sprint 1			
2	1	Project Manager	<p>Planowanie oraz utworzenie backlogu i nakreślenie potrzeb projektowych:</p> <ul style="list-style-type: none">o Utworzenie kroków i backlogu dla DevTeam:<ul style="list-style-type: none">▪ Stworzenie bazy danych i implementacja danych testowych;▪ Weryfikacja bazy danych;▪ Zbudowanie rozwiązania on-premise dla bazy danych;▪ Połączenie szkieletu aplikacji do bazy danych;▪ Funkcjonalność dodawania kont dla użytkowników;▪ Utworzenie podstawowych dostępu dla użytkowników;▪ Uwierzytelniane dwuskładnikowe używając Microsoft Authenticator;▪ Uwierzytelnianie za pomocą odcisku palca;▪ Utworzenie widoków dla danych dostępu (student, dziekanat, wykładowca);▪ Implementacja widoku podstawowego;▪ Implementacja widoku kalendarza;▪ Synchronizacja kalendarza z bazą danych oraz chmurą;▪ Dodawanie wydarzeń do kalendarza;▪ Implementacja oraz zakup licencji na użytek Google Maps w celach biznesowych;▪ Synchronizacja lokalizacji z bazą danych oraz chmurą;▪ Utworzenie oraz implementacja widoku ustawień z funkcjonalnością;▪ Stworzenie powiadomień mailowych;

			<ul style="list-style-type: none"> ▪ Utworzenie powiadomień, gdy użytkownik loguje się z nowego urządzenia; ▪ Optymalizacja aplikacji;
2	1	Project Manager	<p>Planowanie oraz utworzenie backlogu i nakreślenie potrzeb projektowych:</p> <ul style="list-style-type: none"> o Utworzenie kroków i backlogu dla UX Design: <ul style="list-style-type: none"> ▪ Identyfikacja potencjalnych narzędzi do użytku przez designerów; ▪ Stworzenie makiet prototypu: <ul style="list-style-type: none"> • Co najmniej 4 makiety do wyboru przez team oraz Product Ownera i Product Managera; • Przedstawienie początkowego prototypu klientowi; ▪ Korekta prototypu; ▪ Niefunkcjonalne demo dla poszczególnych dostępów: <ul style="list-style-type: none"> • Widok główny; • Widok kalendarza; • Widok ustawień; • Widok powiadomień; • Widok mapy; ▪ Korekta dema; ▪ Implementacja kolorystyki; ▪ Implementacja rozwiązań zespołu programistów: <ul style="list-style-type: none"> • Integracja rozwiązań głównych (widok podstawowy i poszczególnych dostępów); • Integracja mapy; • Integracja powiadomień systemowych; • Integracja powiadomień mailowych; ▪ Optymalizacja aplikacji;
2	1	UX Designer	Identyfikacja potencjalnych narzędzi do użytku przez designerów (w tym: zakup licencji)
2	2	Outsourcing	Outsourcing utworzenia infrastruktury dla deweloperów, deadline stworzenia podstawy: Miesiąc trzeci, tydzień pierwszy (zewnętrzna firma będzie współpracować z zespołem programistów, w celu implementacji testów w ramach pipeline'ów używając dostarczonego rozwiązania chmurowego)
2	2	Developer	Konfiguracja on-permise dla baz danych, utworzenie bazy danych, konfiguracja
2	2	Developer	<p>Identyfikacja potencjalnych narzędzi do użytku przez deweloperów:</p> <ul style="list-style-type: none"> o Rozwiązanie bazodanowe/rozwiązanie serwerowe/dobór framework'ów: <ul style="list-style-type: none"> ▪ Akceptacja ▪ Korekta ▪ Utworzenie początkowej dokumentacji

2	2	Project Manager/ Product Owner/ Developer	Konfiguracja oraz zastosowanie rozwiązań deweloperskich, zakup licencji
2	3	UX Designer	Rozpoczęcie implementacji prototypu interfejsu do wglądu deweloperów
2	3	Developer	Zbudowanie szkieletu aplikacji łącząc się z bazą danych
2	3	Developer	Weryfikacja działania aplikacji i stabilności połączenia
2	3	Developer	Utworzenie testowych kont
2	3	Developer	Testowanie na urządzeniach docelowych
2	4	Project Manager/ Product Owner/ Developer	Akceptacja infrastruktury deweloperskiej oraz zastosowanych narzędzi
2	4	Developer	Migracja na nową infrastrukturę wraz z narzędziami deweloperskimi
2	4	Project Manager/ Product Owner	Sprint review
Sprint 2			
3	1-2	Developer	Finalizacja migracji oraz kontynuowanie developmentu na nowej platformie
3	1-2	Developer	Funkcjonalność dodawania kont dla deweloperów
3	1-2	Developer	Utworzenie podstawowych dostępów dla użytkowników
3	1-2	Developer	Uwierzytelnianie dwuskładnikowe za pomocą Microsoft Authenticator
3	1-2	Developer	Uwierzytelnianie za pomocą odcisku palca
3	3	UX Designer	Korekta prototypu interfejsu
3	3	UX Designer	Utworzenie i finalizacja widoku głównego
3	3	Developer/ UX Designer	Implementacja funkcjonalności widoku głównego
3	4	Outsourcing	Pierwsze testy UX
3	4	Outsourcing	Pierwsze testy bezpieczeństwa dostępu
3	4	Project Manager/ Product Owner	Sprint review
Sprint 3			
4	1	UX Designer	Utworzenie i finalizacja widoku kalendarza
4	1	Developer/ UX Designer	Implementacja funkcjonalności widoku kalendarza
4	2	UX Designer	Korekta widoku kalendarza i/lub widoku głównego
4	2	UX Designer	Dostosowanie widoku kalendarza do poszczególnych dostępów
4	2	Developer/ Outsourcing	Synchronizacja kalendarza z bazą danych i chmurą
4	3	UX Designer	Utworzenie i finalizacja widoku ustawień
4	3	UX Designer	Dostosowanie widoku kalendarza dla poszczególnych dostępów
4	3	Developer	Stworzenie funkcjonalności mapy
4	4	Developer/ UX Designer	Implementacja funkcjonalności widoku ustawień

4	4	Developer	Dodawanie wydarzeń w kalendarzu
4	4	UX Designer	Korekta widoku ustawień
4	4	Project Manager/ Product Owner	Sprint review
Sprint 4			
5	1-2	Developer	Utworzenie podstawowych powiadomień kalendarzowych
5	1-2	Developer	Utworzenie powiadomień, gdy użytkownik loguje się z nowego urządzenia
5	1-2	UX Designer	Utworzenie i finalizacja widoku mapy
5	1-2	UX Designer	Dostosowanie widoku kalendarza dla poszczególnych dostępów
5	3-4	Project Manager/ Product Owner	Weryfikacja funkcjonalności z wytycznymi klienta
5	3-4	UX Designer	Korekta UX
5	3-4	Developer	Bug fixing
5	4	Product Owner/ Project Manager	Sprint review oraz przygotowanie dema rozwiązania
Sprint 5			
6	1	Developer	Optymalizacja funkcjonalności
6	1	UX Designer	Optymalizacja interfejsu
6	2	Product Owner	Prezentacja dema rozwiązania
6	2	Product Owner	Zebranie feedbacku od klienta
6	2	Project Manager/ Product Owner	Planowanie usprawnień
6	3	Product Owner	Przedstawienie planowanych usprawnień klientowi
6	3	Product Owner	Zebranie feedbacku na temat usprawnień
6	3	Product Owner	Ustalenie planu działań z Project Managerem
6	4	Product Owner	Ustalenie możliwości zespołu
6	4	Product Owner	Przedstawienie planu usprawnień klientowi
6	4	Product Owner/ Project Manager	Integracja planu usprawnień do backlogu
7-8	-	Developer	Implementacja usprawnień aplikacji
7-8	-	UX Designer	Implementacja usprawnień interfejsu
9	1-2	Outsourcing	Przeprowadzenie testów automatycznych oraz manualnych
9	1-2	Outsourcing	Przeprowadzenie pen-testów
9	1-2	Outsourcing	Testowanie stabilności infrastruktury oraz bazy danych
9	1-2	Project Manager	Uzupełnienie dokumentacji
9	3-4	Product Owner	Przygotowanie prezentacji projektu
9	4	Project Manager/ Product Owner	Oddanie projektu klientowi

4. Bezpieczeństwo

W celu zapewnienia bezpieczeństwa korzystania z aplikacji dbamy o:

- Zablokowanie zewnętrznego dostępu do serwera. Osoby korzystające z poziomu uprawnień dla dziekanatu nie mają możliwości zalogowania się do aplikacji z innego adresu IP niż uczelnianego;
- W celu zalogowania się do aplikacji zastosowane zostało uwierzytelnianie dwuskładnikowe w formie otrzymanego tokena na numer telefonu podanego podczas tworzenia konta;
- Aplikacja ma regularnie tworzone kopie zapasowe, jest to wykonywane codziennie w godzinach od 02:59 do 3:59. Wybrane godziny zostały celowo, jest to czas kiedy niewielu użytkowników korzysta z aplikacji;
- Zawiadomienia dla każdego użytkownika w formie email. Każde logowanie z nowego urządzenia wysyła wiadomość na email uczelniany alarmujący o zalogowaniu się z nowego urządzenia. W emailu również znajduje się wiadomość, która powoduje wylogowanie ze wszystkich urządzeń i zablokowanie konta. W celu odblokowania użytkownik musi skontaktować się z użytkownikiem na poziomie dziekanatu który może odblokować konto i zresetować hasło;
- Użytkownicy telefonów z czytnikiem linii papilarnych mają możliwość włączenia opcji, która wymaga aby logowanie za każdym razem odbywało się dodatkowo przy pomocy odcisku palca;

5. Opis procesu testowania

Testy aplikacji przeprowadzane są za pośrednictwem wykwalifikowanej firmy outsourcingowej.

- Testy sprawdzające czy aplikacja jest pozbawiona błędów.
- Test godności zaimplementowanej funkcjonalności z dokumentacją.
- Przygotowanie raportu błędów (częstotliwość- 1 raz w tygodniu). Przekazanie raportu osobom odpowiedzialnym za ich naprawę.
- Ponowna weryfikacja błędów w kolejnej wersji aplikacji
- Udostępnienie wersji testowej aplikacji odbiorcom (wybrane osoby z każdej grupy: dziekanat, prowadzący, studenci). Szczegóły testów:
 - Studenci- test prędkości otrzymywanych dla tej grupy powiadomień;
 - Prowadzący - test prędkości wysyłanych powiadomień do wybranych grup studentów, test prędkości otrzymanych powiadomień;
 - Dziekanat – test prędkości powiadomień wysyłanych do prowadzących oraz studentów;
 - Wszyscy: raport z ogólną oceną funkcjonalności oraz przydatności urządzenia; wykaz błędów utrudniających korzystanie z aplikacji. Częstotliwość publikacji raportu – 2 razy w miesiącu (ok. 1WD oraz 11 WD)
- Naprawa wskazanych przez osoby testujące błędów. Maksymalny czas na naprawę błędów- 1 tydzień (5 dni roboczych).
- Proces testowania kończy się w momencie, gdy aplikacja spełnia swoje założenia - ułatwia planowanie czasem użytkowników, ułatwia komunikację oraz organizację sal do prowadzenia zajęć.

6. Protokoły z realizacji wybranych zadań

- Project manager - Patryk 12525

Project manager (również Scrum Master) zarządza podziałem zadań w projekcie oraz odpowiada za efektywność zespołu w procesie tworzenia funkcjonalności. Koordynuje oraz kieruje pracą zespołu w realizacji projektu zgodnie z metodologią Scrum. Jego zadania to między innymi przestrzeganie zasad metodyki Scrum, pomoc w organizacji zespołu oraz planowaniu sprintów.

Jego obowiązkiem jest również rozwiązywanie wszelkich konfliktów i trudności, które mogą wystąpić podczas realizacji projektu. Poza tymi obowiązkami jest odpowiedzialny za rozwój zespołu, ulepszanie komunikacji między jego członkami oraz utrzymanie ciągłości współpracy na dobrych warunkach.

Te obowiązki zostały zrealizowane w tym projekcie, przy tworzeniu dokumentacji, koordynowaniu zadań zespołu, czy też ścisłej współpracy z Product Ownerem przy dostarczeniu finalnego rozwiązania.

- Product owner/ analityk biznesowy - Paulina 13817

Product Owner w metodyce zwinnej Scrum odpowiada za proces tworzenia produktu. Jest to jedyna osoba w projekcie, która odpowiada za wizję oraz wartość docelowego produktu. Stanowi „łącznik” pomiędzy klientem (inwestorem) a osobami zaangażowanymi w projekt. Do jego najważniejszych zadań należy dbanie o klarowną wizję produktu, nadawanie odpowiednich priorytetów, porządkowanie zadań w backlogu. Jest on uczestnikiem wszystkich spotkań scrumowych, dba o postęp prac zgodny z harmonogramem.

Osoba zatrudniona na stanowisku Product Ownera powinna być wysoko wykwalifikowana technicznie. Powinna rozumieć procesy zachodzące w projekcie oraz skomplikowane etapy projektowania.

- Developer - Jakub 13862

Developer odpowiada za stworzenie części kodu, przeprowadzenie testów, bezpieczeństwo aplikacji oraz utworzenie odpowiedniej dokumentacji. Jest odpowiedzialny za utworzenie funkcjonalnych przycisków dla interfejsów użytkowników, stworzenie oraz konfigurację bazy danych, dobranie odpowiedniego szkieletu pod budowę aplikacji, weryfikację działania wraz z optymalizacją i przeprowadzeniem testów. Jego obowiązkiem jest zadbanie o odpowiednie bezpieczeństwo użytkowników którzy będą korzystać z aplikacji. Współpracuje z outsourcingiem w celu synchronizacji oraz naprawy błędów. Jego praca obejmuje stworzenie kalendarza wraz z powiadomieniami. W końcowej fazie dba o optymalizację funkcjonalności i usprawnienie działania aplikacji.

Osoba pełniąca funkcję developera musi być wykwalifikowanym programistą z doświadczeniem pracy w grupie oraz posiadać zakres z umiejętności w obszarze technicznym jak i procesowym.

- UX Design - Patryk 14943
- Full Stack Developer – Emil 15068

Full Stack Developer jest odpowiedzialny za projektowanie, implementację oraz utrzymanie rozwiązania zbudowanego i dostarczonego dla projektu. Jego zadania to między innymi pisanie kodu do aplikacji na warstwie Front-End oraz Back-end. Wspomaga pracę zespołów deweloperskich dzięki swojej obszernej wiedzy i doświadczeniu, dostarcza pomoc i jest pomocną ręką w codziennych problemach procesu tworzenia aplikacji. Jest odpowiedzialny za implementację oraz testowanie aplikacji, integracja z bazą danych czy też tworzenie interfejsu. Bierze udział w Sprintach tworzonych przez Project Managera i pracuje ściśle z każdym zespołem by upewnić się, że tworzone rozwiązanie jest na zaplanowanym poziomie.