**Постановка задачи приложения**

1. **Проблема**

В настоящее время психология стала востребованным для людей и найти хорошего специалиста в этом направлении очень сложно.

В утвержденном порядке оказания психологической помощи и ее стандартах отсутствует требования к профилактике психологических заболеваний, как и у взрослых так и у детских пациентов нуждающиеся на помощь специалистов.

Программа профилактики психологических заболеваний, для пациентов не разработано.

1. **Сценарий**

Севара студентка Ташкентского Межрайонного Университета, изучает информационные технологии, к сожалению она не понимает это направление как остальные его однокурсники, но она старается успевать на все занятия и сделать что требуют преподаватели.

Наконец-то наступил этот час (ИТОГОВЫЫЫЕ). Все преподаватели поназадавали заданий для усвоения пройденных тем, а она не успевала, столько домашних дел, не давали ей нормально заниматься уроками, она хотела учиться но уставала морально и физически потому что была замужем).

Её никто не понимал и она себя тоже… И однажды на переменке как она рассказывает подошла к ней подруга и спросила- “что с ней?”. Севара не знала что ей ответить и тогда её подруга поняла, что с ней что-то нето и прорекомендовала один сайт, где можно найти хороших специалистов своих дел и просто записаться на их курс.

Тогда Севаре было не до этого, но через несколько дней она решила сделать это и зашла на сайт которую рекомендовала ее подруга. Она быстро и легко зарегистрировалась, долго думала на какой курс записаться, потому что все курсы были очень полезными и самое главное всё было по карману.

И в итоге она записалась на 2 психологические курсы потому заинтересовалась и хотела помочь самой себе.

На следующий день она сама прибежала к подруге и обняла ее таак сильно, что подруга испугалась, Севара была так счастлива и благодарна подруге.

1. **Требование**

В проекте должны быть учтены следующие функциональные требования (FR) и нефункциональные требования (NFR).

FR1: Поиск доступных курсов: Студент может просмотреть все курсы текущего семестра по своему основному и второстепенному предмету. Он может присоединиться к курсу, который сохраняет его в своем списке курсов. Он также может отказаться от курса.

FR2: Проверьте сведения о курсе: Студент может просмотреть подробную информацию о курсе, такую как время прохождения курса, расположение лекционного зала на карте и других участников курса, включая их имя и фотографию.

FR3: обновить профиль: Студент может обновить настройки своего профиля и фотографию профиля. Он также может изменить настройки уведомлений.

FR4: Добавление комментариев: Студент может добавлять комментарии к курсу и, таким образом, начинать обсуждение. Другие могут поставить лайк этому комментарию и написать последующие комментарии.

FR5: Запрос на дружбу: Студент может запросить дружбу с другим студентом, который затем получит уведомление о запросе. Второй студент может принять и отклонить дружбу, о чем оба уведомляют первого студента.

FR6: Просмотр курсов друзей: Студент может просматривать курсы своих друзей.

FR7: Просмотр объявлений: Студент может просматривать объявления о курсах и комментировать / ставить им лайки.

FR8: публиковать обновления на временной шкале: Учащийся может публиковать обновления на своей временной шкале. Друзья получают уведомления об обновлениях и могут комментировать их и ставить лайки. Некоторые обновления публикуются автоматически, например, сохранение курса в списке курсов или комментирование курса.

FR9: смотрите календарь курсов: Студент может просматривать все курсы в календаре.

NFR1: Удобство использования: Приложение должно быть интуитивно понятным в использовании, а пользовательский интерфейс должен быть простым для понимания. Все взаимодействия должны быть завершены менее чем за три клика.

NFR2: Соответствие руководящим принципам: Дизайн приложения должен соответствовать рекомендациям по удобству использования для выбранной операционной системы.

NFR3: Целевая платформа: Приложение должно быть разработано на Python.

NFR4: Серверная система: Клиент предоставляет серверную систему с несколькими сервисами, которые необходимо использовать в приложении.

Дополнительные ограничения:

• Система контроля версий должна быть git.

• Документация по исходному коду должна быть в формате HTML.

**4. Целевая среда**

Приложение должно быть продемонстрировано на Python.

**5. Документация**

• Документ с анализом требований (RAD - Requirements Analysis Document)

• Проектный документ системы (SDD - System Design Document)

• Исходный код под контролем версий, включая документацию по исходному коду

**6. Критерии** **приемлемости клиента**

Приложение должно демонстрировать по крайней мере, следующую функциональность: оно показывает свободное время специалистов, к которым пациент может записаться. Пациент может видеть время прибытия до самой клиники. Приложение взаимодействует с северной системой, предоставленной заказчиком, и соответствует требованиям к удобству использования.