Ministerul Educaţiei, Culturii și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Departamentul Ingineria Software și Automatică

**RAPORT**

Lucrare de laborator Nr.6

Disciplina: AMOO

Tema: Realizarea diagramelor de stare și activitate

A efectuat: st.gr. TI-185 Plaschin Andrei

A verificat : lect.univ.   
 Sava Nina  
Melnic Radu

Chișinău 2020

**Scopul**: studierea noțiunilor de stare, stare compusă cu substari concurente, depuse și disjuncte, activitate, condiție de gardă, bară de sincronizare (fork, join), bloc de decizie.

**Sarcina**: de realizat 2 diagrame de stare și 2 diagrame de activitate pentru sistemul informațional ales

**Теория**:Диагра́мма де́ятельности (англ. activity diagram) — UML-диаграмма, на которой показаны действия, состояния которых описано на диаграмме состояний. Под деятельностью (англ. activity) понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов — вложенных видов деятельности и отдельных действий англ. action, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Диаграммы деятельности состоят из ограниченного количества фигур, соединённых стрелками. Основные фигуры:

Прямоугольники с закруглениями — действия

Ромбы — решения

Широкие полосы — начало (разветвление) и окончание (схождение) ветвления действий

Чёрный круг — начало процесса (начальный узел)

Чёрный круг с обводкой — окончание процесса (финальный узел)

Стрелки идут от начала к концу процесса и показывают потоки управления или потоки объектов (данных).

**Диаграмма состояний** — это, по существу, диаграмма состояний из теории автоматов со стандартизированными условными обозначениями, которая может определять множество систем от компьютерных программ до бизнес-процессов. Используются следующие условные обозначения:

* Круг, обозначающий начальное состояние.
* Окружность с маленьким кругом внутри, обозначающая конечное состояние (если есть).
* Скруглённый прямоугольник, обозначающий состояние. Верхушка прямоугольника содержит название состояния. В середине может быть горизонтальная линия, под которой записываются активности, происходящие в данном состоянии.
* Стрелка, обозначающая переход. Название события (если есть), вызывающего переход, отмечается рядом со стрелкой. Охраняющее выражение может быть добавлено перед «/» и заключено в квадратные скобки (***название\_события*[охраняющее\_выражение]**), что значит, что это выражение должно быть истинным, чтобы переход имел место. Если при переходе производится какое-то действие, то оно добавляется после «/» (***название\_события*[охраняющее\_выражение]/действие**).
* Толстая горизонтальная линия с либо множеством входящих линий и одной выходящей, либо одной входящей линией и множеством выходящих. Это обозначает объединение и разветвление соответственно.

**Выполнение работы:**

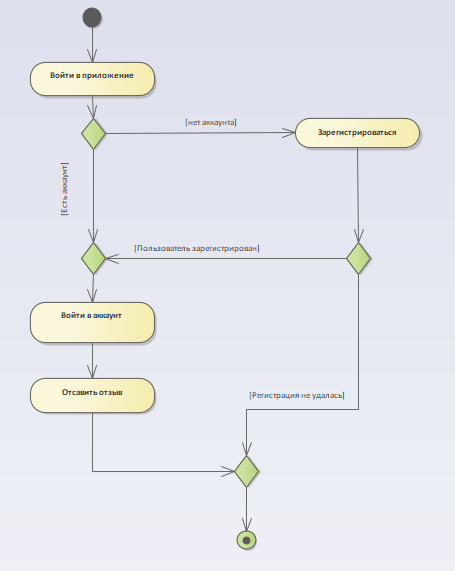


Рис. 1 – Добавление отзыва к приложению

На данной диаграмме показан процесс добавления отзыва от пользователя к приложению в сервисе. Только авторизованный пользователь может оставлять отзыв, в противном случае неавторизованный пользователь должен создать аккаунт и только тогда сможет оставлять отзыв. Если пользователю будет отказано в регистрации, то он не сможет оставить отзыв.

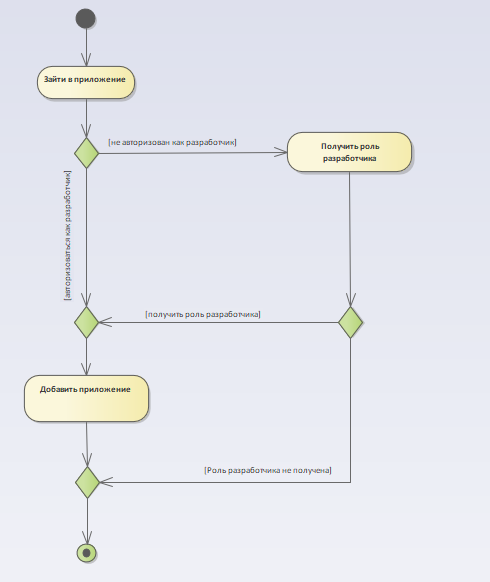


Рис. 2 – Добавление приложения

На данной диаграмме показан процесс добавления приложения со стороны разработчика. В данном сервисе статус разработчика надо приобрести или получить напрямую от админа сервиса. Если пользователь имеет статус аккаунта разработчика, то он напрямую сможет добавлять приложения на сервис.

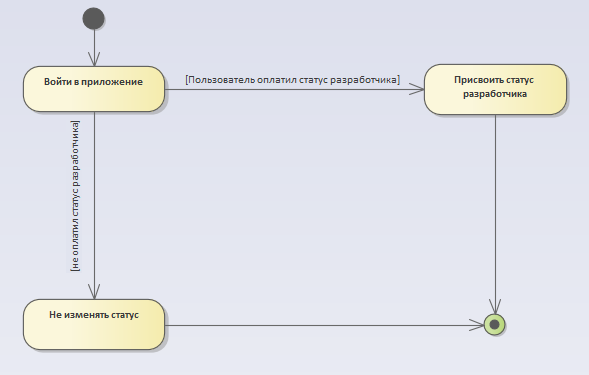


Рис. 3 –Установка статуса/роли разработчика

При регистрации пользователь не имеет роль разработчика, т.е. весь функционал, присущий разработчику, ему не доступен. Если обычный User хочет иметь весь функционал, доступный разработчику, то он должен оплатить лицензию. В ином случае он может продолжить пользоваться системой под статусом пользователя.

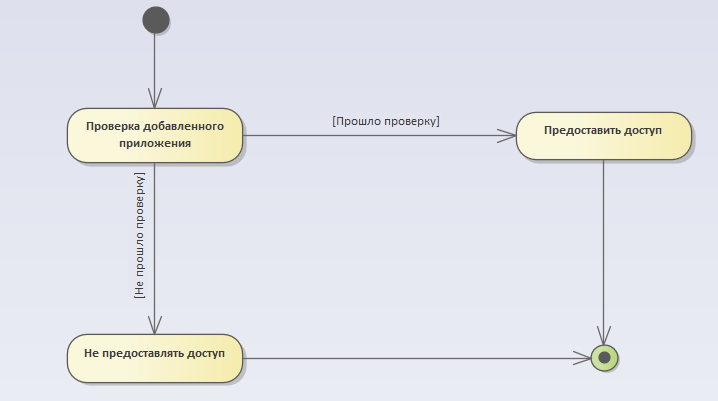


Рис. 4 – Проверка приложения

На диаграмме показан процесс добавки приложения на сервис. После того, как разработчик залил свое приложение, оно не сразу попадает в общий доступ, а должно пройти проверку от администрации сервиса. Это сделано для того, чтоб нежелательный контент не попадал в общий доступ. Так что приложение должно пройти сначала проверку, а только потом оно сможет быть доступно всем пользователям.

**Выводы:**

При выполнения данной лабораторной работы я составил 4 диаграммы, 2 диаграммы деятельности и 2 диаграммы состояния, для выбранной информационной системы, а так же разобрался в построение данных диаграмм.

**Библиография:**

1. **Melnic R., Sava N.** Indrumar metodic “Analiza si modelarea sistemelor informationale”.
2. **Моделирование бизнес процессов|CASE средства|Rational Rose**, [Электронный ресурс].-Режим доступа: <http://www.kpms.ru/Automatization/Rational_Rose.htm>