Андрієць Богдан Валентинович

Use Case Diagram:

* Самі назви Use Cases є доволі прозорими.
* Зв’язок між Use Cases є незрозумілим й на мою думку в деяких місцях неправильним. Автор зв’язує Use Cases за допомогою асоціацій ( association ), я б використав directed.
* Процес пошуку описаний доволі ясно та зрозуміло, але є ще багато функціоналу: можна зайти під своїм gmail account й google search стає чимось більшим. Звичайно я не кажу про опис “Google Office”, Gmail, т.д., але можна було вказати, що взагалі є такі можливості. На мою думку не всі сценарії використання використанні.
* Можна добавити можливість авторизації для User. Або краще створити два актора User та User with google account бо в них різні можливості використання пошукової системи.
* Через відсутність глосарія важко зрозуміти деякі аспекти діаграми, а саме:

Що таке “Add notes”? Можливо малося на увазі “Add bookmark”, але, якщо ні – то для того, щоб створити нотатку потрібний обліковий запис Google.

З діаграми видно, що користувач може шукати тільки нотатки...

Class Diagram:

* Через відсутність глосарія важко зрозуміти деякі аспекти діаграми, а саме: різниця між User та Portal User.
* В Class є User з певною характеристикою (Name, age, email), тобто є можливість авторизації, але в Use Case Diagram цей момент не описано.
* Не зовсім зрозуміло чому зв’язок між Portal User та View Results – generalization та composition одночасно (це взагалі можливо ?).
* Загалом поділ на класи виконано доцільно окрім того моменту, який я описав вище + згідно діаграми результатом пошуку може бути посилання, картинка та карта, тобто гугл не може знайти відео, аудіо-файл, тощо ???
* Зовсім не зрозуміло, що таке hitratio().

Sequence Diagram:

* Важко коментувати через відсутність глосарія.

Висновок:

На мою думку автор в своїх моделях показ не весь функціонал, немає глосарію, діаграми є місцями не зрозумілими. Було використано мало типів діаграм, були використані не всі важливі сценарії. Зв’язок між класами в деяких місцях незрозумілий.

Гончарук Ангеліна Віталіївна

Use Case Diagram:

* Самі назви Use Cases є доволі прозорими (зрозуміло що є що, але не завжди ☺).
* Зв’язок між Use Cases є досить зрозумілим, але є декілька нюансів. Перше – автор показує, що зв’язок між “Check list” та “Subscribed to”, “Check list” та “Member of” – “include”, хоча за логікою повинно бути “extend”.

Activity Diagram Sign In:

* Описано дуже детально, але при виборі “Forget password ?” схема виконання виконана дещо заплутано (важко сприйняти).

Activity Diagram Sign Up, **Activity Diagram Message,** Activity Diagram Tweet, Activity Diagram Search:

* Все лаконічно та зрозуміло.

Висновок:

На мою думку автор в своїх моделях показ не увесь функціонал, немає глосарію, діаграми є зрозумілими, але вони повністю не описують функціонал ( точніше дані діаграми описують тільки user interface). Було використано мало типів діаграм, були використані не всі важливі сценарії. Не було показано зв’язок між класами, тощо.

Пилипець Гліб Володимирович

Use Case Diagram:

* Самі назви Use Cases є доволі прозорими, вони повністю описують предметну область.
* Все описано чітко та зрозуміло.
* Конектори були використанні досить коректно.

Activity Diagrams:

* Описані основні та найбільш важливі сценарії використання.
* Моделі чітко описують поведінку даної системи.
* Конектори були використанні досить коректно.

Class Diagram:

* Самі назви є доволі прозорими, вони повністю описують предметну область.
* Автор чітко зобразив архітектуру системи. Чітко зрозуміло як частини коду функціонують між собою.
* Не дивлячись на великий розмір самої діаграми, через грамотний поділ системи на класи діаграма є доступною для розуміння.
* Зв’язок між класами є зрозумілим.

Висновок:

На мою думку автор показав увесь функціонал та чітко зобразив його. Хочу зазначити, що глосарій не був присутній, але автор робив виноски в самих діаграмах, тому моделі були цілком доступні та зрозумілими. Було використано досить типів діаграм, щоб зрозуміти функціонал та архітектуру системи.

Порівнюючи версії програми “до” та “після” можу сказати що було виконано ряд покращень, які торкнулися не тільки архітектури, а й функціонала загалом.

Печкуров Віталій Михайлович

Через відсутність другої частини, буду перевіряти першу.

Use Case Diagram:

* Краще було б зробити одного актора User ( як я зрозумів Skype для бізнесу не розглядається).
* Деякі прецеденти маю дивні назви, які не описанні в глосарії. Наприклад, “Skype application” ( тобто в Skype є Skype ??? Рекурсія ??? ). Не зрозумілий зв’язок  між Receiver та Skype account (навіщо він ???).

Activity Diagrams:

* Автор намагався на одній діаграмі зобразити увесь функціонал. На мою думку, він пропустив декілька моментів. По перше –  немає можливості зареєструватися, по друге – не врахована можливість листування та надсилання файлів, тощо, по третє – розглянуті не всі можливі сценарії.

Class Diagram:

* Самі назви є доволі прозорими, вони чітко описують предметну область.
* Автор нечітко зобразив архітектуру системи. Не зрозуміло як частини коду функціонують між собою ( мається на увазі зв’язок між класами).
* Не вказані типи зв’язку між класами.

Component Diagram:

* Модель чітко описує поведінку даної системи.
* Конектори були використанні досить коректно.
* Зв’язок між компонентами є зрозумілим.

Sequence Diagram:

* На мою думку Process Call and Process Answer Call можна опустити.
* Модель чітко описує поведінку даної системи.
* Конектори були використанні досить коректно.
* Зв’язок між компонентами є зрозумілим.

## State Diagram, Object Diagram та **Deployment Diagram**:

* Модель чітко описує поведінку даної системи.
* Конектори були використанні досить коректно.
* Зв’язок між компонентами є зрозумілим.

Extra !!!!!

В пріорітеті перевіряти другу частину, якщо її немає – першу.

Добринська Анастасія Олександрівна (Перша частина)

Use Case Diagram:

* Через свій розмір діаграму не можливо сприйняти + при масштабуванні не можливо прочитати назви кейсів.
* Автор розглянув не основний, а увесь функціонал ( цього не потрібно було робити )
* Важко коментувати більш детально, краще взагалі переробити цю діаграму.

Activity Diagram:

* Ті ж самі зауваження та проблеми.

Class Diagram:

* Самі назви є доволі прозорими, вони повністю описують предметну область.
* Автор чітко зобразив архітектуру системи. Чітко зрозуміло як частини коду функціонують між собою.
* Зв’язок між класами є зрозумілим.

Sequence Diagram:

* Модель чітко описує поведінку даної системи.
* Конектори були використанні досить коректно.
* Зв’язок між компонентами є зрозумілим.