

## 스탬프/포인트 시스템 메커니즘

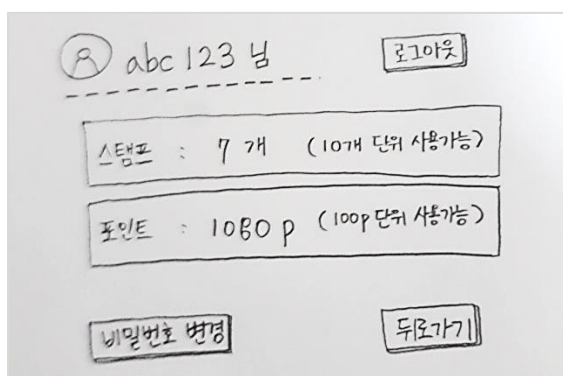
### 1. Requirements

<Customer 모드> (추가사항)

REQ	Type	Pri	Content
188	FR	4	시스템은 Customer 가 마우스 클릭으로 "사용자 정보" 레이블을 선택할 수 있게 해야 한다.
189	FR	4	시스템은 Customer 가 <주문하기> 화면에서 "사용자 정보" 레이블을 클릭했을 경우 비회원 사용자 모드일 때 <Customer 로그인> 화면으로 이동해야 한다.
190	FR	4	시스템은 Customer 가 <주문하기> "사용자 정보" 레이블을 클릭했을 때, Customer 가 로그인 상태일 경우 해당 계정의 <사용자 정보> 화면으로 이동해야 한다.

<사용자 정보>

REQ	Type	Pri	Content
191	FR	3	시스템은 Customer 가 마우스 클릭으로 "뒤로 가기" 버튼을 선택할 수 있게 해야 한다.
192	FR	3	시스템은 Customer 가 "뒤로 가기" 버튼 혹은 스마트폰의 이전 버튼을 선택했을 때 <주문하기> 화면으로 돌아가야 해야 한다.
193	FR	3	시스템은 Customer 가 마우스 클릭으로 "로그아웃" 버튼을 선택할 수 있게 해야 한다.
194	FR	3	시스템은 Customer 가 "로그아웃" 버튼을 선택했을 때 비회원모드로 전환하여 <Customer 모드> 화면으로 돌아가야 한다.
195	FR	3	시스템은 Customer 가 마우스 클릭으로 "비밀번호 변경" 버튼을 선택할 수 있게 해야 한다.
196	FR	3	시스템은 Customer 가 "비밀번호 변경" 버튼을 선택했을 때 로그인 된 상태로 <비밀번호 변경> 화면으로 이동해야 한다.
197	FR	1	시스템은 로그인 된 Customer 계정의 데이터베이스로부터 스탬프 수와 적립된 포인트에 관한 데이터를 불러와 문자열 레이블을 수정하여 보여주어야 한다.
198	NFR	3	시스템은 <사용자 정보> 화면으로 이동했을 때 사용자 계정의 스탬프 개수를 표시하는 문자열 레이블과, 사용자 계정의 적립된 포인트를 표시하는 문자열 레이블, "비밀번호 변경" 버튼, "뒤로 가기" 버튼, "로그아웃" 버튼을 화면에 보여주어야 한다.

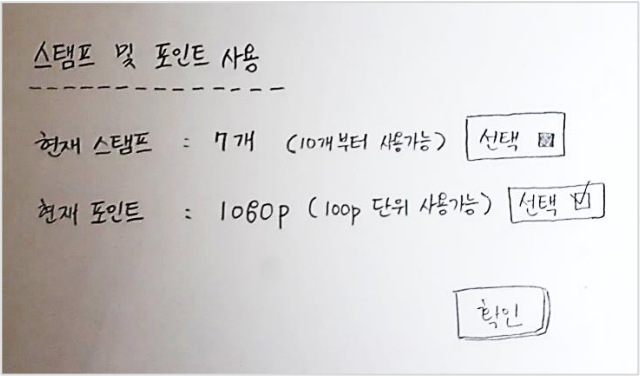
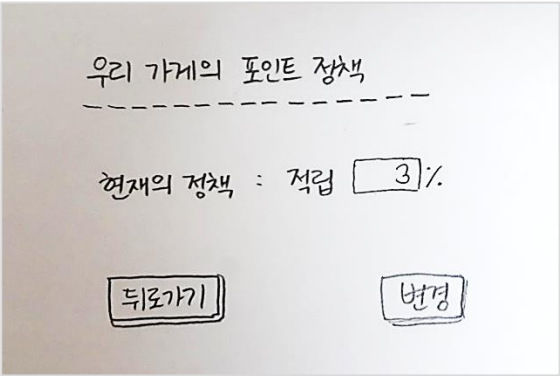


<Admin 모드> (추가사항)

REQ	Type	Pri	Content
199	FR	4	시스템은 Admin 이 마우스 클릭으로 "포인트 정책" 버튼을 선택할 수 있게 해야 한다.
200	FR	4	시스템은 Admin 이 "포인트 정책" 버튼을 클릭했을 때, <포인트 정책> 화면으로 이동해야 한다.

<포인트 정책>

REQ	Type	Pri	Content
201	FR	4	시스템은 Admin 이 마우스의 클릭을 사용하여 "뒤로 가기" 버튼을 선택할 수 있게 허용해야 한다.
202	FR	4	시스템은 Admin 이 "뒤로 가기" 버튼 혹은 스마트폰의 이전 버튼을 선택했을 때 <Customer 모드> 화면으로 돌아가야 한다.
203	FR	3	시스템은 Admin 계정의 데이터베이스로부터 포인트 정책에 관한 데이터를 불러와 해당 값으로 텍스트 필드를 초기화한다.
204	FR	3	시스템은 Admin 이 텍스트필드에 키보드로 숫자를 변경할 수 있게 허용해야 한다.
205	FR	3	시스템은 Admin 이 마우스 클릭으로 "변경" 버튼을 선택할 수 있게 허용해야 한다.
206	FR	3	시스템은 Customer 가 "변경" 버튼을 선택했을 때, 텍스트필드에 입력된 값이 1~100 사이의 숫자인 경우 해당 값을 포인트 정책을 저장하는 데이터베이스에 업데이트 한다.
207	NFR	1	시스템은 <포인트 정책> 화면으로 이동했을 때 숫자를 입력할 수 있는 텍스트필드, "변경" 버튼, "뒤로 가기" 버튼을 화면에 보여주어야 한다.
208	NFR	1	시스템은 1~100 사이의 값이 텍스트 필드에 입력되어 변경 버튼이 선택된 경우 "변경되었습니다!"라는 메시지창을 띄운다.
209	NFR	1	시스템은 1~100 사이가 아닌 값이 텍스트 필드에 입력되어 변경 버튼이 선택된 경우 "1~100 사이의 숫자를 입력해주세요!"라는 메시지창을 띄워야 한다.



### <스탬프 및 포인트 사용>

REQ	Type	Pri	Content
210	FR	4	시스템은 Customer 의 스탬프가 10 개 미만일 때, "스탬프 사용" 체크박스를 비활성화 해야 한다.
220	FR	4	시스템은 Customer의 포인트가 100p 미만일 때, "포인트 사용" 체크박스를 비활성화 해야 한다.
230	FR	3	시스템은 Customer 가 마우스 클릭으로 활성화된 체크박스를 선택할 수 있게 해야 한다.
231	FR	3	시스템은 Customer 가 마우스의 클릭을 사용하여 "확인" 버튼을 선택할 수 있게 해야 한다.
232	FR	3	시스템은 Customer 가 결제확인 메시지창의 확인 버튼을 선택할 경우 포인트 사용 결과를 적용하여 결제금액을 산정하고, 스탬프 및 포인트 사용 결과를 데이터베이스에 업데이트 한 다음 <주문하기> 화면의 "주문완료" 탭으로 이동해야 한다.
233	FR	3	시스템은 Customer 가 결제확인 메시지창의 취소 버튼을 선택할 경우 "결제하기" 버튼을 누르기 전 상태의 <주문하기> 화면으로 돌아가야 한다.
234	NFR	1	시스템은 <스탬프 및 포인트 사용> 화면으로 이동했을 때 Customer 계정의 스탬프 및 포인트 현황을 보여주는 문자열 레이블, "스탬프 사용" 체크박스, "포인트 사용" 체크박스, "확인" 버튼을 화면에 보여주어야 한다.
235	NFR	1	시스템은 "확인" 버튼이 선택되었을 때 "결제를 진행 하시겠습니까?"라는 결제확인 메시지창을 띄워야 한다.

### <포인트 적립하기>

REQ	Type	Pri	Content
236	FR	4	시스템은 Admin 이 마우스 클릭으로 "조리완료" 버튼을 선택할 수 있게 해야 한다.
237	FR	4	시스템은 Admin 이 "조리완료" 버튼을 클릭했을 때, 해당 주문이 유효한 회원의 주문인지 데이터베이스를 통해 확인해야 한다.
238	FR	4	시스템은 해당 주문이 유효한 회원의 주문일 때, 주문 금액과 포인트 정책에 따른 적립예정 포인트를 해당 Customer 의 데이터베이스에 업데이트해야 한다.

### <스탬프 적립하기>

REQ	Type	Pri	Content
239	FR	4	시스템은 Customer 가 마우스 클릭으로 "후기쓰기" 버튼을 선택할 수 있게 해야 한다.
240	FR	4	시스템은 Customer 가 "후기쓰기" 버튼을 클릭했을 때, 해당 회원이 유효한지 데이터베이스를 통해 확인해야 한다.
241	FR	4	시스템은 Customer 가 유효한 회원임을 확인하면, 해당 Customer 의 데이터베이스에 스탬프 개수를 변경하여 업데이트해야 한다.

## 2. Use Cases

Use Case UC- 1: 사용자 정보	
Related requirement	REQ 191~198
Initiating Actor	Customer
Actor's Goal	자신의 계정 상태를 확인하고 관리한다.
Participating Actors	Database, Smartphone
Preconditions	Customer는 스탬프 및 포인트 현황을 Read 할 수는 있으나 write는 불가하다.
Postconditions	None
Flow of Events for Main Success Scenario	
<p>→ 1. Customer는 &lt;주문하기&gt; 화면의 "사용자 정보" 버튼(우측 상단의 아이디 문자열)을 누른다.</p> <p>← 2. 시스템은 로그인 된 유저인지 확인하고, (a) 로그인 상태인 경우 Customer에게 &lt;사용자 정보&gt; 화면으로 넘어가고, (b) 비로그인 상태일 경우 &lt;Customer 로그인&gt; 화면으로 넘어간다.</p> <p>← 3. (a) 시스템은 로그인 된 해당 Customer 계정의 스탬프/포인트 정보를 데이터 베이스에서 불러오고, 해당 데이터의 내용이 담긴 문자열 레이블과 (c)"비밀번호 변경", (d)"로그아웃", (e) "뒤로 가기" 버튼을 보여준다.</p> <p>← 4. (b) 만약 비로그인 상태인 경우 &lt;Customer 로그인&gt; 화면으로 넘어간다.</p> <p>→ 5. (c) Customer는 "비밀번호 변경" 버튼을 누른다.</p> <p>← 6. 시스템은 &lt;비밀번호 변경&gt; 화면으로 넘어간다.</p> <p>→ 5. (d) Customer는 "로그아웃" 버튼을 누른다.</p> <p>← 6. 시스템은 계정을 로그아웃하고 &lt;Customer 모드&gt; 화면으로 넘어간다.</p> <p>→ 5. (e) Customer는 "뒤로 가기" 버튼을 누른다.</p> <p>← 6. 시스템은 &lt;주문하기&gt; 화면으로 돌아간다.</p>	

Use Case UC- 2: 포인트 정책	
Related requirement	REQ 201~209
Initiating Actor	Admin
Actor's Goal	포인트 적립 정책을 결정한다.
Participating Actors	Database, Smartphone
Preconditions	Admin은 주문금액의 몇 %의 포인트를 적립할 것인지 write 할 수 있다.
Postconditions	None
Flow of Events for Main Success Scenario	
<p>→ 1. Admin은 "포인트 정책" 버튼을 누른다.</p> <p>← 2. 시스템은 &lt;포인트 정책&gt; 화면으로 넘어간다.</p> <p>← 3. 시스템은 Database에 저장된 현재의 정책으로 텍스트필드의 값을 초기화한다.</p> <p>→ 4. Admin이 텍스트필드에 1~100 사이의 숫자를 입력하고 "변경" 버튼을 누른다.</p> <p>← 5. "변경" 버튼이 선택되었을 때 텍스트 필드의 숫자가 1~100 사이일 경우 포인트 적립 퍼센티지를 Database에 저장한후 "변경되었습니다!"라는 메시지창을 띄운다.</p> <p>← 6. "변경" 버튼이 선택되었을 때 텍스트 필드의 문자열이 1~100 사이의 숫자가 아닌 경우 "1~100 사이의 숫자를 입력해주세요!"라는 메시지창을 띄운다.</p> <p>→ 7. Admin은 "뒤로 가기" 버튼 혹은 스마트폰의 이전 버튼을 누른다.</p> <p>← 8. 시스템은 &lt;Admin 모드&gt; 화면으로 돌아간다.</p>	

Use Case UC- 3: 스탬프 및 포인트 사용	
Related requirement	REQ 210~235
Initiating Actor	Customer
Actor's Goal	Customer는 결제 전 자신의 스탬프와 포인트를 사용한다.
Participating Actors	Database, Smartphone
Preconditions	Customer는 스탬프 개수와 포인트 현황을 Read하여 사용할 수 있다.
Postconditions	None
Flow of Events for Main Success Scenario	
<p>→ 1. Customer는 &lt;주문하기&gt; 화면에서 "결제하기" 버튼을 누른다.</p> <p>← 2. 시스템은 Customer가 로그인 상태일 경우 &lt;스탬프 및 포인트 사용&gt; 화면으로 넘어간다.</p> <p>← 3. 시스템은 로그인 된 Customer 계정의 스탬프 및 포인트 현황을 화면에 보여준다.</p> <p>← 4. 시스템은 Customer의 스탬프가 10개 미만일 때, "스탬프 사용" 체크박스를 비활성화 한다.</p> <p>← 5. 시스템은 Customer의 포인트가 100p 미만일 때, "포인트 사용" 체크박스를 비활성화 한다.</p> <p>→ 6. Customer는 활성화된 "스탬프 사용" 체크박스 또는 "포인트 사용" 체크박스를 선택한다.</p> <p>→ 7. Customer는 "확인" 버튼을 누른다.</p> <p>← 8. 시스템은 "결제를 진행 하시겠습니까?"라는 확인메시지 창을 띄운다.</p> <p>→ 9. (a) Customer는 확인을 누른다.</p> <p>← 10. "스탬프 사용" 체크박스가 선택상태일 경우 시스템은 "스탬프 10개 사용으로 혜택이 적용됩니다."라는 메시지창을 띄우고 Customer의 스탬프 데이터를 (기존포인트 - 10)으로 변경하여 저장한다.</p> <p>← 11. "포인트 사용" 체크박스가 선택된 상태면서 (c) (포인트 &lt;= 결제금액) 일 때, 시스템은 (기존 포인트 - 기존 포인트를 100으로 나눈 나머지)를 결제금액에서 차감한 후, Customer의 포인트 데이터를 (기존포인트를 100으로 나눈 나머지)로 변경하여 저장한다.</p> <p>← 12. "포인트 사용" 체크박스가 선택된 상태면서 (d) (포인트 &gt; 결제금액) 일 때, 시스템은 결제금액을 0원으로 변경한 후, Customer의 포인트 데이터를 (기존포인트-결제금액)으로 변경하여 저장한다.</p> <p>← 13. 시스템은 &lt;주문하기&gt; 화면의 "주문완료" 탭으로 넘어간다.</p> <p>→ 14. (b) Customer는 취소를 누른다.</p> <p>← 15. 시스템은 "결제하기" 버튼을 누르기 전 상태의 &lt;주문하기&gt; 화면으로 돌아간다.</p>	
Flow of Events for Extensions (Alternate Scenarios)	
<p>→ 1. Customer는 &lt;주문하기&gt; 화면에서 "결제하기" 버튼을 누른다.</p> <p>← 2. 시스템은 Customer가 비회원인 상태일 경우 "현재 비회원입니다. 결제를 진행 하시겠습니까?"라는 확인 메시지 창을 띄운다.</p> <p>→ 3. (a) Customer는 확인을 누른다.</p> <p>← 4. 시스템은 &lt;주문하기&gt; 화면의 "주문완료" 탭으로 넘어간다.</p> <p>→ 5. (b) Customer는 취소를 누른다.</p> <p>← 6. 시스템은 확인 메시지 창을 닫는다.</p>	

Use Case UC- 4: 포인트 적립하기	
Related requirement	REQ 236~238
Initiating Actor	Admin
Actor's Goal	Customer의 주문금액에 따라 포인트를 적립한다.
Participating Actors	Database, Smartphone
Preconditions	1. Admin이 조리 완료 버튼을 눌러서 완전한 거래가 성립되어야 한다. 2. Customer와 Admin은 포인트 현황을 직접 write 할 수 없다.
Postconditions	None
Flow of Events for Main Success Scenario	
→ 1. Admin이 조리 완료를 누른다. ← 2. 시스템은 해당 주문의 Customer계정을 확인한다. ← 3. Database는 Admin이 설정한 포인트 정책에 따라 해당 Customer 계정의 포인트를 업데이트 한다.	

Use Case UC- 5: 스탬프 적립하기	
Related requirement	REQ 239~241
Initiating Actor	Customer
Actor's Goal	Customer의 후기작성에 따라 스탬프를 적립한다.
Participating Actors	Database, Smartphone
Preconditions	1. Customer 계정이 로그인 상태여야 한다. 2. Customer는 스탬프 현황을 직접 수정할 수 없다.
Postconditions	None
Flow of Events for Main Success Scenario	
→ 1. Customer가 로그인 상태에서 <후기작성> 페이지에서 작성하기 버튼을 누른다. ← 2. Database는 해당 Customer 계정의 스탬프 현황을 업데이트 한다.	

### 3. Domain Model

#### Use Case UC- 1: 사용자 정보

Responsibility Description	Type	Concept Name
Use Case와 관련된 모든 컨셉을 조정하고 다른 concept에 작업을 위임한다.	D	Controller
해당 아이디의 스탬프 정보, 포인트 정보를 화면에 보여준다.	D	Page Maker
Database에서 사용자 정보 record를 가져와 Database query를 준비한다.	D	Database Connection
user의 현재 상태(스탬프 현황, 포인트 현황), user가 수행할 수 있는 작업(로그아웃, 비밀번호 변경 버튼)을 보여주는 페이지	K	UserInfoPage (Interface Page)
User의 로그인 상태가 유효한지 체크해준다.	D	login checker
데이터베이스 로그 검색을 위한 검색 매개 변수 지정 양식	K	Search Request

Concept pair	Association description	Association name
Controller ↔ Page Maker	Controller 가 Page Maker 에게 준비된 화면에 대하여 요청을 보내면, 해당 요청에 대한 결과값을 반환한다.	Convey_request
PageMaker ↔ Database Connection	Database Connection 은 PageMaker 에게 저장된 정보를 전달한다.	provide
Page Maker ↔ UserInfoPage	Page Maker 는 userInfo Page 를 prepare 한다.	prepares
Controller ↔ UserInfoPage	Controller 는 userInfo Page 를 post 한다.	post
Search Request ↔ Controller	Controller 가 Search Request 쿼리를 받는다.	receives
Controller ↔ login checker	login checker 는 user 가 로그인 된 상태인지 체크하고 Controller 의 요청에 따라 상태를 변경한다.	Convey_request
Controller ↔ Database Connection	Controller 가 Database Connection 에게 요청을 전달한다.	Convey_request

### Use Case UC- 3: 스탬프 및 포인트 사용

Responsibility Description	Type	Concept Name
Use Case와 관련된 모든 컨셉을 조정하고 다른 concept에 작업을 위임한다.	D	Controller
해당 아이디어의 스탬프 정보, 포인트 정보를 화면에 보여준다.	D	Page Maker
Database의 사용자 정보에 접근하기 위해 Database query를 준비하고, 사용자 정보를 가져오기 위해 Database와 연결한다.	D	Database Connection
user의 현재 상태(스탬프 현황, 포인트 현황), user가 수행할 수 있는 작업(사용 체크박스, 확인 버튼)을 보여주는 페이지	K	UseScorePage (Interface Page)
User의 로그인 상태가 유효한지 체크해준다.	D	login checker
수정된 사용자 정보를 database의 Score 테이블에 저장한다.	D	Archiver
데이터베이스 로그 검색을 위한 요청	K	Search Request
데이터베이스 로그 수정을 위한 요청	K	Update Request
확인을 눌러 포인트 및 스탬프 사용을 완료하면, user에게 Alarm을 준다.	D	Notifier

Concept pair	Association description	Association name
Controller ↔ Page Maker	Controller 가 Page Maker 에게 준비된 화면에 대하여 요청을 보내면, 해당 요청에 대한 결과값을 반환한다.	Convey_request
PageMaker ↔ Database Connection	Database Connection 은 PageMaker 에게 저장된 정보를 전달한다.	provide
Page Maker ↔ useScorePage	Page Maker 는 useScorePage 를 prepare 한다.	prepares
Controller ↔ useScorePage	Controller 는 useScorePage 를 post 한다.	post
Search Request ↔ Controller	Controller 가 Search Request 쿼리를 받는다.	receives
Update Request ↔ Controller	Controller 가 Update Request 쿼리를 받는다.	receives
Controller ↔ login checker	login checker 는 user 가 로그인 된 상태인지 체크하여 Controller 에게 전달한다.	Convey_request
Controller ↔ Database Connection	Controller 가 Database Connection 에게 요청을 전달한다.	Convey_request
Archiver ↔ Database Connection	Archiver 가 Database Connection 에게 데이터 수정 request 에 대한 저장을 요청한다.	Request update
Controller ↔ Notifier	Controller 가 포인트 및 스탬프 사용에 관한 알림을 전달한다.	request notify



## Use Case UC- 2: 포인트 정책

Responsibility Description	Type	Concept Name
Use Case와 관련된 모든 컨셉을 조정하고 다른 concept에 작업을 위임한다.	D	Controller
해당 아이디어의 포인트 정책에 대한 정보를 화면에 보여준다.	D	Page Maker
Database의 사용자 정보에 접근하기 위해 Database query를 준비하고, 사용자 정보를 가져오기 위해 Database와 연결한다.	D	Database Connection
user의 현재 포인트 정책 상태, user가 수행할 수 있는 작업(변경, 뒤로가기 버튼)을 보여주는 페이지	K	PointPolicyPage
User의 로그인 상태가 유효한지 체크해준다.	D	login checker
수정된 사용자 정보를 database의 Score 테이블에 저장한다.	D	Archiver
데이터베이스 로그 검색을 위한 요청	K	Search Request
데이터베이스 로그 수정을 위한 요청	K	Update Request
확인을 눌러 변경을 완료하면, user에게 Alarm을 준다.	D	Notifier

Concept pair	Association description	Association name
Controller ↔ Page Maker	Controller 가 Page Maker 에게 준비된 화면에 대하여 요청을 보내면, 해당 요청에 대한 결과값을 반환한다.	Convey_request
PageMaker ↔ Database Connection	Database Connection 은 PageMaker 에게 저장된 정보를 전달한다.	provide
Page Maker ↔ PointPolicyPage	Page Maker 는 PointPolicyPage 를 prepare 한다.	prepares
Controller ↔ PointPolicyPage	Controller 는 PointPolicyPage 를 post 한다.	post
Search Request ↔ Controller	Controller 가 Search Request 쿼리를 받는다.	receives
Update Request ↔ Controller	Controller 가 Update Request 쿼리를 받는다.	receives
Controller ↔ login checker	login checker 는 user 가 로그인 된 상태인지 체크하여 Controller 에게 전달한다.	Convey_request
Controller ↔ Database Connection	Controller 가 Database Connection 에게 요청을 전달한다.	Convey_request
Archiver ↔ Database Connection	Archiver 가 Database Connection 에게 데이터 수정 request 에 대한 저장을 요청한다.	Request update
Controller ↔ Notifier	Controller 가 포인트 정책 변경에 관한 알림을 전달한다.	request notify

#### Use Case UC- 4: 포인트 적립하기

Responsibility Description	Type	Concept Name
Use Case와 관련된 모든 컨셉을 조정하고 다른 concept에 작업을 위임한다.	D	Controller
Database의 사용자 정보에 접근하기 위해 Database query를 준비하고, 사용자 정보를 가져오기 위해 Database와 연결한다.	D	Database Connection
User의 로그인 상태가 유효한지 체크해준다.	D	login checker
수정된 사용자 정보를 database의 Score 테이블에 저장한다.	D	Archiver
데이터베이스 로그 검색을 위한 요청	K	Search Request
데이터베이스 로그 수정을 위한 요청	K	Update Request

Concept pair	Association description	Association name
Search Request ↔ Controller	Controller 가 Search Request 쿼리를 받는다.	receives
Update Request ↔ Controller	Controller 가 Update Request 쿼리를 받는다.	receives
Controller ↔ login checker	login checker 는 user 가 로그인 된 상태인지 체크하여 Controller 에게 전달한다.	Convey_request
Controller ↔ Database Connection	Controller 가 Database Connection 에게 요청을 전달한다.	Convey_request
Archiver ↔ Database Connection	Archiver 가 Database Connection 에게 데이터 수정 request 에 대한 저장을 요청한다.	Request update

## Use Case UC- 5: 스탬프 적립하기

Responsibility Description	Type	Concept Name
Use Case와 관련된 모든 컨셉을 조정하고 다른 concept에 작업을 위임한다.	D	Controller
Database의 사용자 정보에 접근하기 위해 Database query를 준비하고, 사용자 정보를 가져오기 위해 Database와 연결한다.	D	Database Connection
User의 로그인 상태가 유효한지 체크해준다.	D	login checker
수정된 사용자 정보를 database의 Score 테이블에 저장한다.	D	Archiver
데이터베이스 로그 검색을 위한 요청	K	Search Request
데이터베이스 로그 수정을 위한 요청	K	Update Request

Concept pair	Association description	Association name
Search Request ↔ Controller	Controller 가 Search Request 쿼리를 받는다.	receives
Update Request ↔ Controller	Controller 가 Update Request 쿼리를 받는다.	receives
Controller ↔ login checker	login checker 는 user 가 로그인 된 상태인지 체크하여 Controller 에게 전달한다.	Convey_request
Controller ↔ Database Connection	Controller 가 Database Connection 에게 요청을 전달한다.	Convey_request
Archiver ↔ Database Connection	Archiver 가 Database Connection 에게 데이터 수정 request 에 대한 저장을 요청한다.	Request update