

LBAW 2022/23 Checklists

Each group will have a copy of this spreadsheet. With each artefact presentation, the group must also present the respective self-evaluation checklist filled. The checklist should be updated until the component is submitted.

The checklist for each artefact will be added to the public spreadsheet as an individual sheet. For each artefact delivery, the group should copy the corresponding sheet from the public spreadsheet (link below) to the group's spreadsheet.

Reference Checklists (one checklist per sheet)

https://docs.google.com/spreadsheets/d/1vqQ1vtH-MCd9PhucusyhYy0rvg4hScfZ0xC_-D45-ac

To copy an individual sheet to your document, click the arrow next to the sheet, click on "copy to" then "existing spreadsheet" and select your document.



Checklists are used to help on the development of each artefact.
Mark with '1' the items you worked on, and with '0' the items not done.

Until the component delivery, students can update the corresponding artefacts' checklists.
The "Comments" column can be used to highlight certain aspects of each item.

ER - Requirements Specification			ER	Comments
General	1.1	The component reference and name are clear (e.g. ER - Requirements Specification)	1	
	1.2	The document has a date	1	
	1.3	All authors are identified	1	
	1.4	The editor is identified	1	
	1.5	The document includes a revision history describing the changes (i.e. change log)	1	
	1.6	The name of the file submitted is correct (lbaw2255_er.pdf)	1	
	1.7	The document uses a clear technical writing style	1	
	1.8	There are no spelling, grammar or edition errors	1	
	1.9	There are no references or dependencies of external materials	1	
	1.10	There is no copy&paste text from the examples given	1	
	1.11	The component is updated on the group's wiki	1	
Artefacts	2.1	The A1 checklist is complete	1	
	2.2	The A2 checklist is complete	1	
	2.3	The A3 checklist is complete	1	
	2.4	Group Self-Evaluation is updated	1	

A1. Project Presentation			A1	Comments
Context & Motivation	1.1	The artefact reference and name are clear (e.g. A1 - Project title)	1	
	1.2	The context of the project is briefly presented	1	
	1.3	The project stakeholders are identified	1	
	1.4	The motivation for the project is presented	1	
Goals & Objectives	2.1	The goals and objectives of the system are presented	1	
	2.2	The goals and objectives presented are plausible in this context	1	
Features	3.1	Includes user authentication features	1	
	3.2	Includes user management and administration features	1	
	3.3	Includes advanced search features	1	
	3.4	Includes the core features highlighted in the topic description	1	
	3.5	Includes the core features highlighted in the common description	1	
Access Groups	4.1	All user groups are clearly identified	1	
	4.2	Each user group is briefly described	1	
	4.3	The relation between each user group and the features is clear	1	
	4.4	Includes a user group for administration (user management, ...)	1	
	4.5	Includes a non-authenticated user group (i.e. guests)	1	

A2. Actors and User Stories			A2	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
UML	2.1	The actors diagram is presented	1	
	2.2	Actors are represented using standard UML	1	
	2.3	Generalizations are represented using standard UML	1	
	2.4	Standard graphic elements and practices are used (e.g. no colors, only black & white)	1	
Actors	3.1	Actors are vertically organized	1	
	3.2	Generic actors are at the top, specializations under them	1	
	3.3	Each actor is briefly described	1	
	3.4	Actors represent roles	1	
	3.5	All actors can be mapped to the user roles described in A1	1	
	3.6	Actors clearly map all the user roles defined in A1	1	
	3.7	A non-authenticated actor is defined (the most general)	1	
	3.8	There is a 'Visitor' role for sign-up and login actions	1	
	3.9	An authenticated actor is defined (it is not the 'registered user')	1	
	3.10	An administrator actor is defined	1	
	3.11	If used, external APIs are identified by the service name (e.g. "Google Maps API")	1	OnlyFEUP will not have the necessity of external APIs, this is explained under features on the PDF.
	3.12	The author/owner of the post/comment/bid/etc is defined (depends on the topic)	1	
User Stories	4.1	User stories are organized in one section per actor	1	
	4.2	Each user story has a unique identifier	1	
	4.3	User stories specify roles [who]	1	
	4.4	User stories have capabilities [what]	1	
	4.5	User stories have benefits [why]	1	
	4.6	User stories have priorities	1	
	4.7	User stories are ordered by priority	1	
	4.8	User stories do not overlap	1	
	4.9	User stories have an adequate and consistent granularity	1	
	4.10	User stories support all the features described in A1	1	
	4.11	Login is included	1	
	4.12	Logout is included	1	
	4.13	View/Edit own profile is included	1	
	4.14	Edit/Delete own data (e.g. post/bid)	1	
	4.15	Search posts/questions/tasks/users/etc is included	1	
	4.16	There are user stories to support user management	1	
Supplementary Requirements	5.1	Business rules (BR) are included	1	
	5.2	Includes BR for "can a user vote/comment/review its own stories/answers"	1	
	5.3	Includes BR for "when user is deleted what happens to its content"	1	
	5.4	Includes BR for dates (e.g., exit date >= entry date)	1	
	5.5	Technical requirements (TR) are included	1	
	5.6	Three most critical TR are identified and justified	1	
	5.7	Other restrictions are included	1	

A3. Information Architecture			A3	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
Sitemap	3.1	The sitemap is included	1	
	3.2	Standard notation is used (lines, boxes and stacks of boxes)	1	
	3.3	The sitemap identifies pages, not functions or features	1	
	3.4	Only main links between pages are included	1	
	3.5	Home page is at the top/center	1	
	3.6	Each page has a unique reference	1	
	3.7	Login is presented as a page (may be a page element)	1	
	3.8	Search results page is included	1	
	3.9	About page is included	1	
	3.10	View/Edit own profile is included	1	
	3.11	Administration area and pages are included	1	
	3.12	View project / View question / View post / etc is included	1	
	3.13	View category / View tag / etc is included	1	
Wireframes	4.1	Wireframes are included	1	
	4.2	Basic graphical elements are used (i.e. simple lines, few colors)	1	
	4.3	Wireframes are presented for at least two main screens	1	We have 3
	4.4	For each wireframe, reference zones are indentified	1	
	4.5	Headers and footers are included	1	We have headers and footers features, but they're located in the side bar
	4.6	Navigation structures are included	1	
	4.7	Page titles and headings are included	1	

EBD - Database Specification			EBD	Comments
General	1.1	The component reference and name are clear (e.g. EBD - Database Specification)	1	
	1.2	The document has a date	1	
	1.3	All authors are identified	1	
	1.4	The editor is identified	1	
	1.5	The document includes a revision history describing the changes (i.e. change log)	1	
	1.6	The name of the file is correct (lbawYYgg_ebd.pdf)	1	
	1.7	The document uses a clear technical writing style	1	
	1.8	There are no spelling, grammar or edition errors	1	Not that we know of
	1.9	There are no references or dependencies of external materials	1	
	1.10	There is no copy&paste text from the examples given	1	
	1.11	The component is updated on the group's wiki	1	
	1.12	The production database (at db.fe.up.pt) is updated	1	
	1.13	The source code is updated in the group's Gitlab repository	1	
	1.14	A tag "EBD" is defined in the GitLab repository	1	
Artefacts	2.1	The A4 checklist is complete	1	
	2.2	The A5 checklist is complete	1	
	2.3	The A6 checklist is complete	1	
	2.4	Group Self-Evaluation is updated	1	

A4. Conceptual Data Model			A4	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
UML	2.1	UML notation is consistently used	1	
	2.2	Diagram layout is clear (visual organization)	1	
	2.3	Classes are correctly represented	1	
	2.4	Generalizations are correctly represented	1	
	2.5	Associations are correctly represented	1	
	2.6	Restrictions and business rules are correctly represented	1	
Classes	3.1	Classes are presented with areas for name and attributes	1	
	3.2	Classes do not have methods associated	1	
	3.3	The classes support all high and medium priority user stories defined in A2	1	
	3.4	Class names are consistent (e.g. always singular, always in English)	1	
Associations, multiplicity, roles	4.1	Automatic primary keys are not presented	1	
	4.2	Natural keys are not used as primary keys (e.g. NIF)	1	
	4.3	Multiplicity is defined for all associations	1	
	4.4	Roles are used to explain how an object participates in the relationship	1	
	4.5	Mandatory associations (not null) are indicated in the multiplicity	1	
	4.6	In 1-1 associations, directionality is defined	1	
	4.7	The associations support all high and medium priority user stories defined in A2	1	
	4.8	There is an "Authorship" association	1	
Attributes, domains and restrictions	5.1	All attributes have a generic type (text, number, date, boolean)	1	
	5.2	Attribute visibility is omitted (e.g. '+' prefix not included)	1	
	5.3	Domains are defined for attributes that have predefined fixed values	1	
	5.4	Not null attributes are indicated in the restrictions	1	
	5.5	Unique attributes are indicated in the restrictions	1	
	5.6	Restrictions related to numerical attributes are indicated (e.g. > 0)	1	
	5.7	Restrictions related to date types are indicated (e.g. > today)	1	
	5.8	Attributes with default values are indicated	1	
	5.9	All generalizations have constraints defined	1	
	5.10	All restrictions and business rules defined in A2 are included	1	

A5. Relational Schema			A5	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
Schema	2.1	The compact notation is correctly used	1	
	2.2	Each relation has a unique reference	1	
	2.3	Relation names are lowercase and in snake_case when needed	1	
	2.4	All UML classes are mapped	1	
	2.5	All class attributes are mapped	1	
	2.6	All associations are mapped	1	
	2.7	All relations have a PK	1	
	2.8	No natural keys are used as PK	1	
	2.9	All FK attributes reference a relation	1	
	2.10	In 1-1 associations, a FK is used considering the directionality	1	
	2.11	In 1-N associations, a FK is used	1	
	2.12	In N-N associations, a relation is defined with a composite PK of two FKs	1	
	2.13	Generalisations are correctly mapped and the choices well justified	1	
	2.14	Domains are defined and used if necessary	1	
Integrity rules	3.1	All NN attribute restrictions are included	1	
	3.2	All UK attribute restrictions are included	1	
	3.3	All date attributes have adequate restrictions	1	
	3.4	All numeric attributes have adequate restrictions	1	
Schema validation	4.1	Schema validation section is included	1	
	4.2	For each relation, all candidate keys are listed	1	
	4.3	For each relation, all FD are listed	1	
	4.4	Each relation's normal form is identified	1	
	4.5	The schema's normal form is identified and a justification is provided	1	
	5.1	The SQL script is included	1	
	5.2	The SQL script contains the creation statements	1	
	5.3	The SQL script cleans up the current database state	1	
	5.4	The SQL script is cleaned (e.g. excluded from export comments)	1	
	5.5	Code highlighting is used for readability	1	
	5.6	All domains are included in the SQL script	1	
	5.7	All relations are included in the SQL script	1	
	5.8	PK are defined as SERIAL	1	
	5.9	FK are not defined as SERIAL	1	

SQL Code	5.10	The SQL script works without errors	1	
	5.11	SQL script is included in the group's repository	1	
	5.12	The production database (at db.fe.up.pt) has been set up with the SQL script	1	

A6. Indexes, Integrity and Populated Database			A6	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
Workload	2.1	The workload section is included	1	
	2.2	The relations' magnitude and growth estimation section is included	1	
	2.3	For each relation, magnitude and growth is estimated	1	
Indexes	3.1	Performance indexes are proposed	1	
	3.2	For each index, a relation and attribute(s) is defined	1	
	3.3	For each index, the type is defined	1	
	3.4	For each index, the cardinality is defined	1	
	3.5	For each index, clustering is defined	1	
	3.6	Full-text search (FTS) indexes over multiple fields are proposed	1	
	3.7	For FTS indexes, field weighting is used	1	
	3.8	For each index, a justification is provided	1	
	3.9	For each index, the SQL code is included	1	
	3.10	Indexes are not proposed for PK	1	
	3.11	Indexed are not proposed for UK	1	
Triggers and User Defined Functions	4.1	Triggers and functions are proposed	1	
	4.2	Restrictions not yet covered in the schema are defined for high and medium priority US	1	
	4.3	For each trigger, a justification is included	1	
	4.4	For each trigger, the SQL code is included	1	
Database transactions	5.1	Database transactions section is included	1	
	5.2	Each transaction has an isolation type defined and justified	1	
	5.3	Each transaction has a justification	1	
	5.4	Transactions' SQL syntax is correct	1	
	5.5	No unnecessary transactions are included	1	
	5.6	All transactions for high and medium priority users stories are included	1	
SQL	6.1	The SQL schema script is included	1	
	6.2	The SQL script resets the database state (includes DROPs + CREATES)	1	
	6.3	The SQL schema script executes without errors	1	
	6.4	The SQL population script is included	1	
	6.5	The SQL population script is included in the group's repository	1	
	6.6	The SQL population script executes without errors	1	
	6.7	The SQL schema script is included in the group's repository	1	
	6.8	The production database (at db.fe.up.pt) has been updated with the SQL scripts	1	The production database rejects any queries before a transaction. To run the populate script, all the inserts must be done first, then all the transactions

EAP - Architecture Specification and Prototype		EAP	Comments	
General	1.1	The component reference and name are clear (e.g. EAP - Architecture Specification and Prototype)	1	
	1.2	The document has a date	1	
	1.3	All authors are identified	1	
	1.4	The editor is identified	1	
	1.5	The document includes a revision history describing the changes (i.e. change log)	1	
	1.6	The name of the file is correct (lbawYYgg_eap.pdf)	1	
	1.7	The document uses a clear technical writing style	1	
	1.8	There are no spelling, grammar or edition errors	1	
	1.9	There are no references or dependencies of external materials	1	
	1.10	There is no copy&paste text from the examples given	1	
	1.11	The component is updated on the group's wiki	1	
	1.12	The prototype is updated and working at the production machine	1	
	1.13	The source code is updated in the group's Gitlab repository	1	
	1.14	A tag "EAP" is defined in the GitLab repository	1	
Artefacts	2.1	The A7 checklist is complete	1	
	2.2	The A8 checklist is complete	1	
	2.3	Group self-evaluation is updated	1	

A7. Web Resources Specification			A7	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
Modules	2.1	The Modules section is included	1	
	2.2	Each module has a unique identifier	1	
	2.3	Each module has a name and a brief description	1	
	2.4	Modules granularity is adequate	1	
Permissions	3.1	The Permissions section is included	1	
	3.2	Each permission has a unique identifier	1	
	3.3	Each permission has a name and a brief description	1	
	3.4	Permissions are adequate for the actors defined	1	
	3.5	A 'self'/owner' permission is defined	1	
	3.6	A 'public' permission is defined	1	Named visitor
OpenAPI	4.1	OpenAPI specification in YAML is included	1	
	4.2	Link to OpenAPI specification file (a7_openapi.yaml) in the repository	1	
	4.3	Specification is consistent in the wiki and in the repository	1	
Web resources (verify for each resource)	5.1	Each web resource has a unique identifier	1	
	5.2	Path (URL) is defined	1	
	5.3	HTTP method is defined	1	
	5.4	Each resource has a unique (URL, HTTP method) pair	1	
	5.5	Permissions are defined	1	
	5.6	URL identify resources not actions (sometimes actions on resources)	1	
	5.7	View (GET) resources have UIs associated	1	
	5.8	Action (PUT/POST) resources have success and error redirects	1	
	5.9	Action (PUT/POST) resources do not have an UI	1	
	5.10	API resources do not have an UI	1	
	5.11	API resources have return codes and sometimes return data (e.g. JSON)	1	
	5.12	All arguments have mandatory or optional indications	1	
	5.13	All arguments have a type defined	1	
	5.14	All web resources for high priority user stories are included	1	

A8. Vertical Prototype			A8	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
Implemented features	2.1	Implemented features section is included	1	
	2.2	Authentication is implemented	1	
	2.3	Logout is implemented	1	
	2.4	Features marked as high priority for the vertical prototype are implemented	1	
	2.5	References to the implemented user stories are included	1	
	2.6	Access features are implemented	1	
	2.7	Creation features are implemented	1	
	2.8	Update features are implemented	1	
	2.9	Delete features are implemented	1	
	2.10	AJAX and API features are implemented	1	
	2.11	Permissions control using Policies is implemented	1	
	2.12	Feedback messages (e.g. errors) are implemented	1	
Code quality	3.1	Source code repository is updated	1	
	3.3	The LBAW framework is used	1	
	3.4	All non-essential LBAW template code was removed (e.g. Thingy! code)	1	
	3.5	No additional libraries or tools are used	1	
	3.6	Laravel routes are correctly used	1	
	3.7	Laravel controllers are correctly used	1	
	3.8	Laravel templates are correctly used	1	
	3.9	Laravel data access is correctly used	1	
	Prototype access	4.1	Prototype URL is included	1
4.2		Prototype is online and working at the production machine	1	
4.3		User credentials for testing are provided	1	

A9. Product			A9	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
Product information	2.1	List of implemented user stories are included	1	
	2.2	All user stories have team members associated	1	
	2.3	All user stories have completion state information (as a percentage)	1	
	2.4	Revision history section with changes to A2 and A7 is included	1	
	2.5	Link to the final release source code is included	1	
	2.6	Docker command to start the image from GitLab Container Registry is included	1	
	2.7	URL to the product at the production machine is included	1	
	2.8	Credentials for different roles are included	1	
	2.9	"README.md" is updated with the Docker command, URL and credentials	1	
Features	3.1	Contextual help is supported	1	
	3.2	About page is implemented	1	
	3.3	Input validation on the client (frontend) is implemented	1	
	3.4	Input validation on the server (backend) is implemented	1	
	3.5	All high and medium priority user stories are implemented	1	
	3.6	User authentication is implemented	1	
	3.7	User registration is implemented	1	
	3.8	User account deletion is implemented	1	
	3.9	Password recovery is implemented	1	
	3.10	User access control is implemented	1	
	3.11	Administration features are implemented	1	
	3.12	Full-text search is implemented	1	
	3.13	Full-text search with multiple weighted fields is implemented	1	
	3.14	Advanced search using filters is implemented	1	As the teacher agreed, filters were considered to be the separation of the contents sought by four sections: users, groups, comments and posts
	3.15	AJAX interactions are implemented	1	
	3.16	Feedback messages (e.g. errors, notifications) are implemented	1	
Architecture and technologies	4.1	OpenAPI specification in YAML for the final product is included	1	
	4.2	All non-essential LBAW template code was removed	1	
	4.3	List of additional libraries or packages used is included	1	
	4.4	Non-authorized libraries or packages are not used	1	
	4.5	Complex code blocks are explained with comments	1	
	4.6	Unnecessary code removed (e.g. commented code, debug code)	1	
	4.7	No temporary files are left on the final product	1	
	4.8	Laravel routes are correctly used	1	
	4.9	Laravel controllers are correctly used	1	
	4.10	Laravel templates are correctly used	1	
	4.11	Laravel data access is correctly used	1	
	4.12	Laravel policies are correctly used	1	
	4.13	HTML validation report is included	1	Here the warnings and errors are the result of the bootstrap framework
	4.14	CSS validation report is included	1	
Product usability and accessibility	4.15	HTML semantic elements are used	1	
	4.16	Minimal redundancy in CSS used (i.e. use cascading features)	1	
	4.17	Product not vulnerable to URL manipulation	1	
	4.18	Product not vulnerable to user input manipulation	1	
	4.19	Internal errors are handled	1	
	4.20	Laravel APP_DEBUG variable is set to TRUE in .env	1	
	5.1	Usability checklist report is included	1	
5.2	Accessibility checklist report is included	1		
5.3	Responsive design is implemented	1		

A10. Presentation			A10	Comments
Artefact	1.1	The artefact reference and name are clear	1	
	1.2	The goal of the artefact is briefly presented (1, 2 sentences)	1	
Materials	2.1	Brief product presentation is included	1	
	2.2	URL to the product is included	1	
	2.3	URL to the video is included	1	
	2.4	Video includes textual or audio descriptions	1	
	2.5	Product data is realistic (e.g. no 'lorem ipsum' style)	1	
Presentation	3.1	Application is readily available in production	1	
	3.2	The presentation is well structured	1	
	3.3	Communication is clear	1	
	3.4	The time limit is not exceeded	1	
	3.5	Main features are shown	1	
	3.6	No bugs nor unexpected fails happen	1	
	3.7	Answers during the discussion are clear	1	
Features Presented	4.1	Authentication & Authorisation	1	
	4.2	CRUD operations	1	
	4.3	User input validation on client (HTML, JS)	1	
	4.4	User input validation on server	1	
	4.5	User account creation	1	
	4.6	User account deletion	1	
	4.7	Password recovery	1	
	4.8	Help features (contextual and others)	1	
	4.9	AJAX calls	1	
	4.10	Full-text search	1	
	4.11	User administration	1	

PA - Product and Presentation			PA	Comments
General	1.1	The component reference and name are clear (e.g. PA - Product and Presentation)	1	
	1.2	The document has a date	1	
	1.3	All authors are identified	1	
	1.4	The editor is identified	1	
	1.5	The document includes a revision history describing the changes (i.e. change log)	1	
	1.6	The name of the file is correct (lbawYYgg_pa.pdf)	1	
	1.7	The document uses a clear technical writing style	1	
	1.8	There are no spelling, grammar or edition errors	1	
	1.9	There are no references or dependencies of external materials	1	
	1.10	There is no copy&paste text from the examples given	1	
	1.11	The component is updated on the group's wiki	1	
	1.12	The prototype is updated and working at the production machine	1	
	1.13	The source code is updated in the group's Gitlab repository	1	
	1.14	A tag "PA" is defined in the GitLab repository	1	
Artefacts	2.1	The A9 checklist is complete	1	
	2.2	The A10 checklist is complete	1	
	2.3	Group Self-Evaluation is updated	1	

